

Project Documentation

This document is mainly about the different implementations made in our project. Before securing the cloud infrastructure, we will first be required to create our own website and database. The website created is a simple website named Qaziwa Books with an organised user interface for browsing and purchasing books. The database created will contain all the books available in our store and customer's (registered users) credentials used to register for an account.

Creation of Website:

We have designed a simple website (After the security implementation) to ensure that the functionality works.

Our website consists of:

- login
- register
- logoff
- listing (guest, registered user, administrator)
- shopping cart
- payment
- successful payment
- advertisement (only at cart page)
- verification

Login page:

We started by designing the base of the website which is the login page and registration. The login page code was written where it asks for the user's input which is the email address and password. After the required field is entered and the user clicks on the submit button which is the login button will then take the 2 fields and send them to doLogin.php. It then checks if the field written is similar to any users in the database. This is done by writing a SELECT statement that retrieves the user's details that is the same as the file written. If the field is correct, the page is then redirected to the listing page (with href). Since we wanted to allow the admin rights to edit the listing, we wrote a code that if the user logs in with the 'admin' account it would redirect to a separate listing page that has the ability to delete, add and edit books into the library and from there with a SQL statement, it would update the database accordingly. For users who would log in as a guest, we coded a submit type that redirects the user to the listing page for guests, allowing them to view the books.

QaZiWa Books - Login



Login

Email Address:

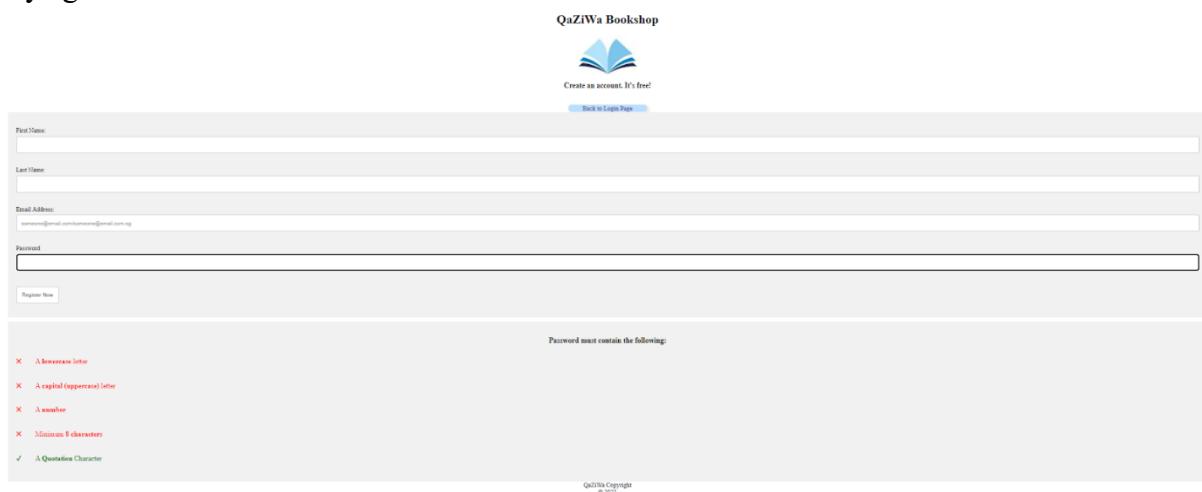
Password:

[Sign-up for free!!](#)

QaZiWa Copyright
© 2021

Register page:

For new users who wish to register an account, the registration button would redirect them to the registration page (register.php) from the elseif line of code stated earlier. For the registration page, we would implement the code to create fields for them to create an account. The respective inputs would then be transferred to doRegister.php with form action. In doRegister.php the first name, last name, email address, and password that has been filled by the user would be inserted into the database. We would use \$_POST to retrieve the fields and write an SQL statement to insert the fields into the individual values. Each field that is posted would then be inputted into the function stripslashes() and RemoveSpecialChar() to prevent any SQL injection statements. Afterward, the code would run an if-else statement where if the insertion is successful, it would href back to the login page (login.php). However, if the insertion was not successful, it would echo “Your account creation was unsuccessful. Please try again.”.



QaZiWa Bookshop

Create an account. It's free!

[Back to Login Page](#)

First Name:

Last Name:

Email Address: commoner@gmail.com/commoner@mail.com.ng

Password:

Password must contain the following:

- A lowercase letter
- A capital (uppercase) letter
- A number
- Minimum 8 characters
- A Quotation Character

QaZiWa Copyright
© 2021

As shown above, we want to make the password strong, therefore we implement validation to the created password field, this is the step of adding the security in which to prevent an attacker from brute force the password. As shown above, we can see that the user needs to

put a lowercase letter, uppercase letter, a number, and a minimum of 8 digit characters. Therefore, in calculation numbers have 10, alphabet (uppercase) have 26 and lower case has 26, and a total of 8 characters. So by calculating this

Password specs	Results
Elements	To crack your password it will take (at max):
<input checked="" type="checkbox"/> Alpha (lower case) <input checked="" type="checkbox"/> Alpha (UPPER case) <input checked="" type="checkbox"/> Numbers <input type="checkbox"/> Symbols	60,650,029.33 hours
Length 8 characters long	2,527,084.56 days
Time * 1 ms between each attempt	6,923.52 years
<input type="button" value="Calculate"/>	

It would roughly be 6,923.52 years and assume 1ms for each attempt. Hence this is the enhanced security on the user side to ensure that their password is not being brute-forced.

Logoff page:

For the logoff page, we start by destroying the user's session by implementing `session_destroy()`. A string "You have logged out." is then given to a variable message to allow the code to echo it on the logout page. We then created the page to look like a logout page with a simple title called "QaZiWa Bookstore Logout".

QaZiWa Bookstore - Logout

You have logged out.

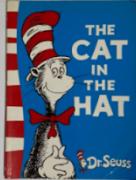
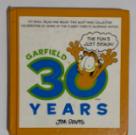
QaZiWa Copyright
© 2021

Listing Page (Guest):

The listing page for users that logs in as a guest has the least privileges. They can only view the books inside the store after creating a guest account.

Welcome to QaZiWa Bookshop - Guest Login

Login You are using guest account. To buy book do Login or Register your account

	Title: The Cat In The Hat Summary: Poor Dick and Sally. It's cold and wet and they're stuck in the house with nothing to do . . . until a giant cat in a hat shows up, transforming the dull day into a madcap adventure and almost wrecking the place in the process! ★★★★★ Rating:5/5 Author: Dr. Suess Price: \$13.57 Quantity: 7 books left
	Title: Garfield 30 Years Summary: 30 years of laugh and lasagna bundled into one book. A tribute to the furry feline after reaching the three-decade milestone this lavishly illustrated volume features more than four hundred strips, including thirty of Jim Davis's all-time favorites panels into one. ★★★★★ Rating:4/5 Author: Jim Davis Price: \$8.16

Listing Page (Registered User):

These users have more privileges as compared to the guest user. Apart from just viewing the books available inside the store, they can make purchases by adding and removing items from a shopping cart called \$arrCart[]. It will be reflected inside the cart_books table inside the database. After confirming their order, they will be directed to the payment page for payment. A successful payment will bring them to the successful payment page and show the estimated time of their delivery. Users can choose to return to the listing page.

Welcome to QaZiWa Bookshop - User Login

Logoff Welcome Mary Sue to QaZiWa Bookstore

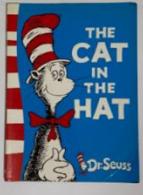
	Title: The Cat In The Hat Summary: Poor Dick and Sally. It's cold and wet and they're stuck in the house with nothing to do . . . until a giant cat in a hat shows up, transforming the dull day into a madcap adventure and almost wrecking the place in the process! ★★★★★ Rating:5/5 Author: Dr. Suess Price: \$13.57 Quantity: 7 books left Add to Cart
	Title: Garfield 30 Years Summary: 30 years of laugh and lasagna bundled into one book. A tribute to the furry feline after reaching the three-decade milestone this lavishly illustrated volume features more than four hundred strips, including thirty of Jim Davis's all-time favorites panels into one.

Listing Page (Administrator):

They have the most privilege among the 3 different types of users. Some basic privileges are adding, deleting, and editing books. All data will be stored inside the database QaZiWa. To add books, an INSERT statement will add them to the database where title_id gives a unique identity to each book (primary key). When editing listed books, the UPDATE statement is used to amend information about books available inside the store. When deleting listed books, a SQL query DELETE FROM statement to delete books from the database.

Welcome to QaZiWa Bookshop - Admin Login

[Logoff](#) [Delete Books](#) [Add Books](#) [Edit Books](#) You are login as Admin


Title: The Cat In The Hat
Summary: Poor Dick and Sally. It's cold and wet and they're stuck in the house with nothing to do . . . until a giant cat in a hat shows up, transforming the dull day into a madcap adventure and almost wrecking the place in the process!
★★★★★ Rating:5/5
Author: Dr. Suess
Price: \$13.57
Quantity:7 books left


Title: Garfield 30 Years
Summary: 30 years of laugh and lasagna bundled into one book. A tribute to the fury feline after reaching the three-decade milestone this lavishly illustrated volume features more than four hundred strips, including thirty of Jim Davis's all-time favorites panels into one.
★★★★★ Rating:4/5

→ Edit books:

[← Back](#) You are login as Admin - This is a Edit page


Title: Beano Annual 2007
Summary: One of British's longest running children's comic this edition compiles all of Beano's escapades in 2007 into one volume.
★★★★★ Rating:3/5
Author: Nigel Auchterlounie
Price: \$5.00
Quantity:10 books left
[Click here to Edit](#)


Title: The Cat In The Hat

→ Add books:

[← Back](#) You are login as Admin - This is a Add book page

Title:
Author:
Summary:
Rating:
Price:
Amount \$
0
Quantity:
Book Image (Front-view):
 No file chosen

→ Delete books:

QaZiWa Bookshop - Delete Books

[← Back](#) You are login as Admin - This is a deletion page


Title: Beano Annual 2007
Summary: One of British's longest running children's comic this edition compiles all of Beano's escapades in 2007 into one volume.
★★★★★ Rating:3/5
Author: Nigel Auchterlounie
Price: \$5.00
Quantity:10 books left


Title: The Cat In The Hat
Summary: Poor Dick and Sally. It's cold and wet and they're stuck in the house with nothing to do . . . until a giant cat in a hat shows up, transforming the dull day into a madcap adventure and almost wrecking the place in the process!
★★★★★ Rating:5/5
Author: Dr. Suess
Price: \$13.57
Quantity:10 books left

Confirm Deletion?

Beano Annual 2007

Shopping Cart Page:

Items that will be added or deleted from the cart are called \$arrCart[] which is the parameter that would get the information from the database accordingly. That will then be assigned to individual parameters which will be used for the CSS styling to tidy up the display of the respective information. This is what makes the quantity, prices, title look organized. For situations where the users suddenly decide to delete the books from their cart, the code which forms the docartdelete.php would be posted for the commands there to run.

For the docartdelete.php the DELETE SQL statement is written there to delete any selected book from the cart_books table. An if-else statement is written to print an error message in the event the delete is unsuccessful.

Going back to the cart.php the cart would access the user's cart information from the database however the information has to be inserted first before any retrieval is possible therefore this is where doAddToCard comes in. The code in this PHP takes the title of the ID of books that the current user is selecting and takes in all the information of that book and inserts it in the cart_books table with the INSERT INTO statement.

QaZiWa Bookshop - View Cart

The screenshot shows a web page titled "Qaziwa Bookshop - View Cart". At the top left is a "Back" button and a message "Missing something? Press back". The main content area displays a single item in the cart:

Book Image	Book Title	Book Price	Deletion Button
	Garfield 30 Years	\$8.16	Delete

To the right of the cart is a sidebar titled "Qaziwa Recommendations" with two sponsored sections:

- Sponsored by Qaziwa Bookstore**: A thumbnail image of a woman holding a book, with the caption "Who is this?" and a link "Why 'Who Is This?' Is Literally The Most Insulting Test Ever (Video)".
- Healthy Morning Drinks to Start Your Day Off**: A thumbnail image of several colorful bottles of vitamin water, with a link below it.

Payment Page

The payment page would be page redirected after the user confirms the books that they would like to purchase at the cart. With that in mind, the code to return to the cart is implemented at the start to allow the user to return to the cart if they wished to do so. We would then add the fields for the user to enter their billing address. Each field has a placeholder to give an example of what each field should look like.

← Back	View Cart - Press back		
Billing Address		Payment	
Full Name <input type="text" value="Andy Lau"/>		Name on Card <input type="text" value="Andy Lau Yan Boon"/>	
Email Address <input type="text" value="john@example.com"/>		Credit card number <input type="text" value="1111-2222-3333-4444"/>	
Residential Address <input type="text" value="542 Hougang Ave 8"/>		Expiry Month/Year <input type="text" value="MM/YY"/>	CVV <input type="text" value="711"/>
Unit Number <input type="text" value="#12-345"/>	Postal Code <input type="text" value="521147"/>		
<input checked="" type="checkbox"/> Shipping address same as billing			
Continue to checkout			
Total Books Cart 1 \$8.16 Garfield 30 Years <hr/> Total \$8.16			

Successful Payment

For the successful payment page, it would appear after the payment page has been successfully redirected after confirming the page. A SQL query is implemented here to delete the current user's cart information from the cart_books table. After the query has been successfully executed, a message will be echoed confirming successful payment with an ETA of the order arrival. A href is also implemented at the end of the message to allow the user to be redirected back to the listing page.



Payment Successful

Thank you for shopping with us!

Your shipment will arrive around 3 weeks.

[Back to Listing page](#)
QaZiWa Copyright
© 2021

Advertisement (Cart Page)

Taken from an open-source page we took the code and its assets and inserted them into our own .js file. Afterward, we added the code to display it on the right side of the cart page. At the end of the few lines would be the code that randomizes all these advertisements and with the function, getRandomInt would return a specific number of advertisements. It would then be targeted to the cart page.

```

1  function (window, undefined) {
2    /* Options */
3
4    var creatives = [
5      {"title": "Is Donald Trump The Best Candidate for 2016? Vote Here.", "creative": "/creatives/trump.jpg", "href": "http://newsmax.com"},  

6      {"title": "These Photos From The Past Are Bitter Sweet", "creative": "/creatives/rifle.jpg", "href": "http://bzulamp.com"},  

7      {"title": "Most Satisfied People Don't Wait For What They Want, They Go Get It", "creative": "/creatives/satisfied.jpg", "href": "http://elitedaily.com/life/"}  

8      {"title": "10 Tips To Learn Any Language From The Genius Who Speaks It", "creative": "/creatives/genius.jpg", "href": "http://elitedaily.com/life/satisfied-pr"},  

9      {"title": "A Lioness Captures A Baby Baboon And Does The LAST Thing You'd Expect", "creative": "/creatives/lioness.jpg", "href": "http://www.littledudha.com/"}  

10     {"title": "Why 'Who Is This?' Is Literally The Most Insulting Tees Ever (Video)", "creative": "/creatives/whois.jpg", "href": "http://elitedaily.com/humor/wh"},  

11     {"title": "The Six Worst Types of Coworkers: And How To Deal With Them", "creative": "/creatives/coworker.jpg", "href": "http://elitedaily.com/"},  

12     {"title": "4 in 5 Americans Are Ignoring Buffet's Warning", "creative": "/creatives/buffet.jpg", "href": "http://www.fool.com/video-alert/stock-advisor/sa-n"},  

13     {"title": "Warren Buffet Just Gave Americans A Big Warning", "creative": "/creatives/buffet.jpg", "href": "http://www.fool.com/video-alert/stock-advisor/sa-n"},  

14     {"title": "Americans Urged To Search Their Name On New Site", "creative": "/creatives/american.jpg", "href": "http://www.instantcheckmate.com/"}  

15     {"title": "The Most Addicted Shopping Site For Women", "creative": "/creatives/addicted.jpg", "href": "http://tophatter.com/blog"},  

16     {"title": "Power Companies Fear This Chicago Family", "creative": "/creatives/solar.jpg", "href": "http://www.thedailylife.com/new-policy-makes-power-compani"},  

17     {"title": "Six Reasons Your Wife Is Your Most Important Relationship", "creative": "/creatives/wif.jpg", "href": "http://elitedaily.com/life/culture/6-reason"},  

18     {"title": "7 Overhyped Games That Ended Up Being Terrible", "creative": "/creatives/terrible.jpg", "href": "http://www.looper.com/323/7-hyped-games-ended-terr"},  

19     {"title": "Games That Punish You Seriously For Dying", "creative": "/creatives/dying.jpg", "href": "http://www.looper.com/3117/games-seriously-punish-dying"},  

20     {"title": "Healthy Morning Drinks To Start Your Day Off", "creative": "/creatives/morningdrinks.png", "href": "http://ephphany.com/site/post/577/"},  

21     {"title": "Best Ways to Scare Your Girlfriend", "creative": "/creatives/scaregirlfriend.png", "href": "http://ephphany.com/site/post/116/"},  

22     {"title": "Movies That Ruined the Book", "creative": "/creatives/booksmovies.png", "href": "http://ephphany.com/site/post/3"},  

23     {"title": "The 10 Most Disturbing Pokemon Facts", "creative": "/creatives/pokemon-facts.png", "href": "http://gamerant.com/pokemon-disturbing-facts"},  

24     {"title": "How To Impress Employers at Info Sessions", "creative": "/creatives/infosessions.png", "href": "http://prteamman.github.io/How-To-Impress/"}  

25     {"title": "So You Want to Date an Artist", "creative": "/creatives/saveartist.png", "href": "http://blog.newhive.com/so-u-want-to-date-an-artist/"}  

26     {"title": "So You Want to Date an Artist", "creative": "/creatives/saveartist.png", "href": "http://blog.newhive.com/so-u-want-to-date-an-artist/"}  

27     {"title": "Weight-lifting Kangaroo Has Neighborhood on High Alert", "creative": "/creatives/buffkang.png", "href": "http://www.theweathernetwork.com/us/news/a"},  

28     {"title": "Are Lentils a Low-Calorie Food?", "creative": "/creatives/lentils.png", "href": "http://healthyeating.sfgate.com/lentils-lowcalorie-food-1034.html"},  

29     {"title": "10 Small Dogs and What They're Like", "creative": "/creatives/pug.jpg", "href": "http://www.sheknows.com/pets-and-animals/articles/808120/top-10-s"},  

30   ];
31
32   /* most recently parsed script */
33   var scripts = document.getElementsByTagName("script");
34   var index = scripts.length - 1;
35   var kaboodleTag = scripts[index];
36   var src = kaboodleTag.getAttribute("src");
37   var kaboodleRoot = src.substring(0, src.lastIndexOf("/"));
38   var kaboodleStyle = document.createElement("link");
39   kaboodleStyle.setAttribute("href", kaboodleRoot + "/kaboodle.css");
40   kaboodleStyle.setAttribute("rel", "stylesheet");
41   kaboodleStyle.setAttribute("type", "text/css");
42   document.head.appendChild(kaboodleStyle);

```

Sponsored by QaZiWa Bookstore



[Is Donald Trump The Best Candidate for 2016? Vote Here.](#)



[The Most Addicted Shopping Site For Women](#)



[So You Want to Date an Artist](#)

[Verification Page](#)

On this page, there will be an OTP that will send over to the user via email, from there the user will then log in to their own email and get the OTP number and key it into the verification page where it will then check if the number tally with the database. If it tallies, then the user is able to use the account and if not, the user will be unable to log into their account.

The screenshot shows a web-based OTP verification interface. At the top is a modal window titled "OTP Verification" with an "OTP" input field and a blue "Submit OTP" button. Below the modal is a message: "Do check your email for your OTP number". Underneath that is copyright information: "QaZiWa Copyright © 2021". Below the page content is a horizontal line with an "Important" section indicator. To the right of the line is a social media post from "fypc300qaziwa" with 31 likes. The post text reads: "OTP Verification - Your OTP verification code is 92051". Below this is another post from "me" with the text "TSS".

This is only the first security testing, but going further than this, we will be using google authenticator to authenticate our website. After intensive research and testing, we have come up with the google authenticator and integrate it into our website. This can be seen below:

The screenshot shows a web-based OTP verification interface. It features a QR code for Google Authenticator, instructions to "Do install Google Authenticator App in your phone and scan the QR code above. After scanning, key in the number code that you see in the App. Thank you :)", and copyright information: "QaZiWa Copyright © 2021". To the right of the page is a photograph of a smartphone displaying the Google Authenticator app. The app screen shows a QR code and the text "QaziwaAuthentication (Gezwe) 007 630".

After scanning the QR code, using the google authenticator app, an OTP number is shown in which then we will key in the number in the web OTP section. This is an added security in which using a reliable source and API from Google to authenticate the OTP, as compared to the first authentication, where the attacker can brute force the OTP due to the first one being using the static OTP and the OTP having no change or time out. But when using the Google authenticator, it has a timeout session and the OTP is not static and keeps changing every few seconds, therefore, making brute force not possible. In addition, the QR code will keep changing into a new QRcode and it will knot be a static QRcode, this is an added security feature to ensure that different users have different access to the QR code and ensure that the OTP pin is kept consistently changing together with the QR code.

Creation of database tables:

Since we already know what the website consists of, now is the time for the database creation. We have planned to use phpMyAdmin as the service for our database. We created a new database from scratch with the name QaZiWa that uses the latin1_swedish_ci collation. Below show the tables that we used for our database.

Books (table):

The table “books” would contain the information of each book that is being sold on QaZiWa. We started by creating the primary key title_id with the auto-increment setting activated for it. The type for the primary key is then set to int with a max count of 11 characters. Next would be creating a column for the title and author individually and what the name suggests would store exactly that with the type being varchar with a maximum number of characters of 40. A summary column is also created with a max character of 8000 for the admin to add in a sizable introduction to the book. Next would be the price which uses the decimal type and the range 5,2 to allow exact pricing down to the cents. The quantity is then added with the int type like the primary key. Lastly would be the picture_front column which uses the type varchar which has a max length of 70. This column keeps the filename of the image of the book which allows the website’s code to easily access the necessary picture of the targeted book. With the data, fields set we then began inserting 11 different books ranging from different genres, ages, and types.



The screenshot shows the phpMyAdmin interface with the 'books' table selected. The left sidebar lists databases: mydb, mysql, oldqazawa, performance_schema, phpmysadmin, qazawa, New, books, cart_books, users, test, verification. The 'books' table structure is shown with columns: title_id, title, author, summary, rating, price, quantity, picture_front. Data rows are listed with titles like 'The Cat In The Hat', 'Garfield 30 Years', etc., and authors like Dr. Seuss, Jim Davis, Russel Lee, Jeff Kinney, Stephen King, Boey, Jeff Kinney, etc. The 'picture_front' column contains file names like 'cat_hat_front.jpg', 'garfield_front.jpg', etc. At the bottom, there are buttons for Check all, With selected, Edit, Copy, Delete, Export, Show all, Number of rows (25), Filter rows, Search this table, Sort by key (None).

	+ Options	title_id	title	author	summary	rating	price	quantity	picture_front
<input type="checkbox"/>	Edit Copy Delete	2	The Cat In The Hat	Dr. Suess	Poor Dick and Sally. It's cold and wet and they're...	5	13.57	8	cat_hat_front.jpg
<input type="checkbox"/>	Edit Copy Delete	3	Garfield 30 Years	Jim Davis	30 years of laugh and lasagna bundled into one boo...	4	8.16	44	garfield_front.jpg
<input type="checkbox"/>	Edit Copy Delete	4	True Singapore Ghost Stories 11	Russel Lee	Russel Lee investigates witchcraft and uncovers it...	2	10.59	7	ghost_stories_front.jpg
<input type="checkbox"/>	Edit Copy Delete	5	Diary of a Wimpy Kid Hard Luck	Jeff Kinney	Greg Heffley's on a losing streak. His best friend...	4	14.93	0	hard_luck_front.jpg
<input type="checkbox"/>	Edit Copy Delete	6	Insomnia	Stephen King	You'll lose a lot of sleep. Ralph does. At first h...	3	16.95	14	insomnia_front.jpg
<input type="checkbox"/>	Edit Copy Delete	7	When I was a kid 2	Boey	A continuation of the hit book "When I Was a Kid" ...	3	19.24	4	kid_front.jpg
<input type="checkbox"/>	Edit Copy Delete	8	Diary of a Wimpy Kid Long Haul	Jeff Kinney	A family road trip is supposed to be a lot of fun ...	4	20.87	20	long_haul_front.jpg
<input type="checkbox"/>	Edit Copy Delete	9	Diary of a Wimpy Kid Old School	Jeff Kinney	Life was better in the old days. Or was it? That's...	2	20.87	13	old_school_front.jpg
<input type="checkbox"/>	Edit Copy Delete	10	As I was Passing II	Adibah Amin	Adibah Amin, the celebrated chronicler of everyday...	3	19.90	3	passing_front.jpg
<input type="checkbox"/>	Edit Copy Delete	11	Transformers Saga of the Altspark	Simon Furman	Saga of the Altspark reveals untold events from th...	5	17.99	25	transformers_front.jpg

Users (table):

The “users” table stores the user’s email/username and password to allow registered users to login into their account. The primary key for this table would be user_id with the same set of auto-increment and typeset just like the books table. We then added columns for their first and last name to allow a bit of personalization when the user logs in. The email column is then added with the varchar typeset and the character count of 40. Lastly would be inserting the password with the type set to varchar and the count of 32. The max character is set to 32 as we then set the password type to be encrypted to an MD5 hashed password as an added layer of security. With all the columns created we then began inserting a few user accounts to test the functionality of the website and also created an admin account that would have the ability to add new books, delete books and edit them from the website itself. In addition, there is also an OTP section where when the user is logged into the website, an OTP will send via their email and from there it will then tally with the current database OTP, and if it tallies the user is successfully logged into the account if not then he or she cannot log into the account.

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	user_id	int(11)	latin1_general_ci		No	None	AUTO_INCREMENT		Change Drop More
2	first_name	varchar(40)	latin1_general_ci		No	None			Change Drop More
3	last_name	varchar(40)	latin1_general_ci		No	None			Change Drop More
4	email	varchar(40)	latin1_general_ci		No	None			Change Drop More
5	md5password	varchar(32)	latin1_general_ci		No	None			Change Drop More
6	otp	varchar(10)	latin1_swedish_ci		No	None			Change Drop More

Indexes

Cart books (table):

Lastly would be the table that saves every individual user's choices of books which are called "cart_books". The primary key of this table would be called cart_id with the auto-increment option activated for it too. Unlike the other 2 tables, "cart_books" would contain a foreign key as there would be a need to access data from the other 2 tables. The foreign key would be the user_id and title_id. From there the other columns created for this table which are the cart_title, cart_price and cart_picture would all be accessed through the foreign key respectively.

Table structure

	cart_id	user_id	title_id	cart_title	cart_price	cart_picture
	3	9	2	The Cat In The Hat	13.57	cat_hat_front.jpg
	4	9	2	The Cat In The Hat	13.57	cat_hat_front.jpg
	5	9	3	Garfield 30 Years	8.16	garfield_front.jpg
	6	9	5	Diary of a Wimpy Kid Hard Luck	14.93	hard_luck_front.jpg
	7	9	5	Diary of a Wimpy Kid Hard Luck	14.93	hard_luck_front.jpg
	8	9	5	Diary of a Wimpy Kid Hard Luck	14.93	hard_luck_front.jpg
	9	9	5	Diary of a Wimpy Kid Hard Luck	14.93	hard_luck_front.jpg
	10	9	5	Diary of a Wimpy Kid Hard Luck	14.93	hard_luck_front.jpg

Creation of cloud networks:

Before we proceed to implement the database and web server onto the instance, we need to create our own public and private network, Virtual Private Cloud (VPC), an internet gateway, route table

Creation of VPC:

So to start off we first need to create a VPC that helps us to have a public and private facing network in which it is suitable for our web (public-facing) & database (private facing) servers.

VPC > Your VPCs > vpc-090a8ad552bdc7ed6

vpc-090a8ad552bdc7ed6 / QaZiWa VPC

Actions ▾

Details Info			
VPC ID vpc-090a8ad552bdc7ed6	State Available	DNS hostnames Enabled	DNS resolution Enabled
Tenancy Default	DHCP options set dopt-a72614dd	Main route table rtb-073c5b0a953c1945f / QaZiWa Route Table	Main network ACL acl-0b0b4f76a0e828b49
Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -	IPv6 CIDR (Network border group) -
Route 53 Resolver DNS Firewall rule groups Failed to load rule groups	Owner ID 501144450828		

Your VPCs (1/2) [Info](#)

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR (Network border group)
vpc-090a8ad552bdc7ed6	vpc-090a8ad552bdc7ed6	Available	10.0.0.0/16	-

IPv4 CIDRs [Info](#)

CIDR	Status
10.0.0.0/16	Associated

IPv6 CIDRs [Info](#)

CIDR	Pool	Status
You have no IPv6 CIDR blocks associated with your VPC.		

For the VPC, we have assigned an IP address range of 10.0.0.0/16 and leave the Tenancy as default. Tenancy defines how EC2 instances are distributed across physical hardware and affect pricing. After we have set up the VPC then we will be continuing to create a subnet

Creation of Subnets:

Firstly, we create a private subnet in which it is for our database server to be placed within the internal network. We have configured the private subnet as shown below.

Subnet ID subnet-04e9d1ba0d738e7d6	State Available	VPC vpc-090a8ad552bdc7ed6 QaZiWa VPC	IPv4 CIDR 10.0.2.0/24
Available IPv4 addresses 250	IPv6 CIDR -	Availability Zone us-east-1a	Availability Zone ID use1-az4
Network border group us-east-1	Route table rtb-0856a597fe2623a4d Qaziwa Route Table for NAT	Network ACL acl-0b0b4f76a0e828b49	Default subnet No
Auto-assign public IPv4 address No	Auto-assign IPv6 address No	Auto-assign customer-owned IPv4 address No	Customer-owned IPv4 pool -
Outpost ID -	Owner 501144450828	Subnet ARN arn:aws:ec2:us-east-1:501144450828:subnet/cuback	

This subnet will not have an internet access/connection, therefore any instances will not be able to connect to the internet unless a NAT service is being implemented in which this configuration will be shown soon.

After creating the private subnet, now is the time to create a public subnet. We will be needing 2 public subnets which are named Qaziwa public subnet 1 & Qaziwa public subnet 2. We need 2 public subnets because later on, we need to configure the load balancer in which it needs 2 zones to ensure that we can configure the load balancer easily and provide scalability.

Qaziwa public subnet 1:

Name	Subnet ID	State	VPC	IPv4 CIDR	IP
<input checked="" type="checkbox"/> Qaziwa public subnet	subnet-0bccaa6218d2bed2e5	Available	vpc-090a8ad552bcd7ed6 QaziWa VPC	10.0.1.0/24	-
<input type="checkbox"/> -	subnet-6bc58334	Available	vpc-cb9a0bb6	172.31.32.0/20	-

Subnet ID subnet-0bccaa6218d2bed2e5	State Available	VPC vpc-090a8ad552bcd7ed6 QaziWa VPC	IPv4 CIDR 10.0.1.0/24
Available IPv4 addresses 248	IPv6 CIDR -	Availability Zone us-east-1a	Availability Zone ID use1-az4
Network border group us-east-1	Route table rtb-073c5b0a953c1945f QaziWa Route Table for Internet Gateway	Network ACL acl-0b0b4f76a0e828b49	Default subnet No
Auto-assign public IPv4 address Yes	Auto-assign IPv6 address No	Auto-assign customer-owned IPv4 address No	Customer-owned IPv4 pool -
Outpost ID -	Owner 501144450828	Subnet ARN arn:aws:ec2:us-east-1:501144450828:subnet-0bccaa6218d2bed2e5	

Qaziwa public subnet 2:

<input type="checkbox"/> -	subnet-cac5fac4	Available	vpc-cb9a0bb6	172.31.64.0/20	-
<input checked="" type="checkbox"/> Qaziwa public subnet 2	subnet-0d563aaca151ca8f8	Available	vpc-090a8ad552bcd7ed6 QaziWa VPC	10.0.3.0/24	-
<input type="checkbox"/> -	subnet-45da5274	Available	vpc-cb9a0bb6	172.31.48.0/20	-

Subnet ID subnet-0d563aaca151ca8f8	State Available	VPC vpc-090a8ad552bcd7ed6 QaziWa VPC	IPv4 CIDR 10.0.3.0/24
Available IPv4 addresses 249	IPv6 CIDR -	Availability Zone us-east-1b	Availability Zone ID use1-az6
Network border group us-east-1	Route table rtb-073c5b0a953c1945f QaziWa Route Table for Internet Gateway	Network ACL acl-0b0b4f76a0e828b49	Default subnet No
Auto-assign public IPv4 address No	Auto-assign IPv6 address No	Auto-assign customer-owned IPv4 address No	Customer-owned IPv4 pool -
Outpost ID -	Owner 501144450828	Subnet ARN arn:aws:ec2:us-east-1:501144450828:subnet-0d563aaca151ca8f8	

With the public subnet, we are able to connect to the internet and the internet can also connect to us. As such this will be good for our web server. Now we will continue to configure the routing table for our public subnet.

Creation of Route table:

This is a routing table that connects to the public subnet and which also enables both of them to be able to ping each other and connects to one another, in addition, this route table is connected to the internet gateway which in other terms connects to the internet.

Route table to internet gateway:

The screenshot shows the AWS Route Tables page. At the top, a table lists route tables by Name, Route table ID, Explicit subnet associations, Edge associations, Main status, and VPC. One row is selected: "QaZiWa Route Table for Internet Gateway" (rtb-073c5b0a953c1945f), which has 2 subnets associated and is set as the Main route table for the vpc-090a8ad55 VPC.

Below the table, a detailed view of the selected route table is shown. It includes fields for Route table ID (rtb-073c5b0a953c1945f), Main status (Yes), VPC (vpc-090a8ad552bdc7ed6 | QaZiWa VPC), and Owner ID (501144450828). The "Details" tab is active, showing explicit subnet associations for 2 subnets.

A secondary table at the bottom shows all route tables in the account, with one row selected: "QaZiWa Route Table for Internet Gateway" (rtb-073c5b0a953c1945f) and another row for "rtb-0cc1a3a9cbab7e493".

The main configuration view for the selected route table (rtb-073c5b0a953c1945f) is displayed. It includes tabs for Details, Routes, Subnet associations, Edge associations, Route propagation, and Tags. The "Subnet associations" tab is active, showing explicit subnet associations for two subnets: "subnet-0bccaa6218d2bed2e5 / Qaziwa public subnet" (IPv4 CIDR 10.0.1.0/24) and "subnet-0d563aaca151ca8f8 / Qaziwa public subnet 2" (IPv4 CIDR 10.0.3.0/24).

Route table to NAT instance:

Due to the private subnet being unable to connect to the internet on its own, therefore, we have come to the conclusion to create a NAT instance that can help to route the traffic out to the internet. So in the configuration above we can see that the routes are looking to the NAT instance which we have already created and this NAT instance will be helping to let the traffic out to the internet but not the other way. We choose NAT instance instead of NAT gateway due to the limitation and fewer privileges of our account in AWS education.

The screenshot shows the AWS Route Tables page. A table lists route tables by Name, Route table ID, Explicit subnet associations, Edge associations, Main status, and VPC. One row is selected: "Qaziwa Route Table for NAT" (rtb-0856a597fe2623a4d), which has 1 subnet associated and is set as the Main route table for the vpc-090a8ad55 VPC.

Below the table, a detailed view of the selected route table is shown. It includes fields for Route table ID (rtb-0856a597fe2623a4d), Main status (No), VPC (vpc-090a8ad552bdc7ed6 | QaZiWa VPC), and Owner ID (501144450828). The "Details" tab is active, showing explicit subnet associations for 1 subnet: "subnet-04e9d1ba0d738e7d6 / Qaziwa private subnet".

The screenshot shows the AWS Route Tables management interface. At the top, there is a search bar labeled "Filter route tables". Below it, a table lists route tables:

	Name	Route table ID	State	Propagated	Owner
<input checked="" type="checkbox"/>	-	rtb-0cc1a3a9cbab7e493	-	-	Yes vpc-022432902
<input checked="" type="checkbox"/>	Qaziwa Route Table for NAT	rtb-0856a597fe2623a4d	subnet-04e9d1ba0d738...	-	No vpc-090a8ad55

Below the table, a message says "rtb-0856a597fe2623a4d / Qaziwa Route Table for NAT". There are tabs for "Details", "Routes", "Subnet associations" (which is selected), "Edge associations", "Route propagation", and "Tags".

The "Subnet associations" section shows one association:

Explicit subnet associations (1)	
<input checked="" type="checkbox"/>	Find subnet association
Subnet ID	IPv4 CIDR
subnet-04e9d1ba0d738e7d6 / Qaziwa private subnet	10.0.2.0/24
IPv6 CIDR	-

At the bottom, there is a "Route tables (1/4) Info" section with a "Create route table" button.

Below the route table list, another "Subnet associations" section is shown:

Route tables (1/4) Info	
<input checked="" type="checkbox"/>	rtb-0cc1a3a9cbab7e493
<input checked="" type="checkbox"/>	Qaziwa Route Table for NAT
rtb-0856a597fe2623a4d	subnet-04e9d1ba0d738...
-	-
Yes vpc-022432902	No vpc-090a8ad55

At the bottom, there is a "Routes (2)" section with a "Edit routes" button.

After we have successfully created our route tables, now is the time to create internet gateways.

Creation of Internet Gateways:

Internet gateway has made use to connect to the internet in which it will be useful for our web server to provide the website to all of the users.

The screenshot shows the AWS Internet Gateways management interface. At the top, there is a search bar labeled "Filter internet gateways". Below it, a table lists internet gateways:

	Name	Internet gateway ID	State	VPC ID	Owner
<input checked="" type="checkbox"/>	Qaziwa gateway	igw-01a83952e83d2c8df	Attached	vpc-090a8ad552bdc7ed6 QaZiWa VPC	501144450828
<input type="checkbox"/>	-	igw-a2e521d8	Attached	vpc-cb9a0bb6	501144450828

Below the table, a message says "igw-01a83952e83d2c8df / Qaziwa gateway". There are tabs for "Details" (which is selected) and "Tags".

The "Details" section shows the following information:

Details			
Internet gateway ID	State	VPC ID	Owner
igw-01a83952e83d2c8df	Attached	vpc-090a8ad552bdc7ed6 QaZiWa VPC	501144450828

Creation of cloud and online available services:

Since we have already settled the cloud networking in which we have created the VPC, private and public subnet, route table and internet gateway, now is the time to create and implement the various services and configuration that we have planned and what AWS services have offered to us. We will be using and configuring the Instances, Application Load Balancer, Auto scaling, Snapshots, AMI, Volumes, CloudFlare, InnoDB encryption, IAM, KMS, CloudWatch, CloudTrail, Security group and many more.

Creation of S3 Bucket Storage - Website/Database:

In the AWS cloud, we started by creating a new bucket in the S3 table. We called it “qaziwabucket”. Setting the initial settings for the S3 bucket we deactivated the checked setting of the bucket blocking all public access to allow us to access the file publicly. After removing the block settings we then uploaded a zip file containing all the codes for our website. After successful uploading, we then made the zip file public to allow our web server instance to access it.

The screenshot shows the AWS S3 console interface. At the top, there's an 'Account snapshot' section with metrics like Total storage (2.8 MB), Object count (2), and Avg. object size (1.4 MB). Below this is a 'Buckets' section with a table showing one bucket named 'qaziwabucket'. The table includes columns for Name, AWS Region, Access, and Creation date. The 'Access' column for the bucket indicates 'Objects can be public'. At the bottom, the 'Objects' section is shown with a table listing two objects: 'C300.zip' (zip type, June 16, 2021, 13:51:33 (UTC+08:00), 2.8 MB, Standard storage class) and 'qaziwa.sql' (sql type, June 2, 2021, 17:33:48 (UTC+08:00), 8.4 KB, Standard storage class). The 'Actions' dropdown menu is visible for both objects.

This is where we can use the wget method to fetch this zip file so that we can easily implement the website source code into the AWS instance web server and it also applies for the AWS instance database server. Once we have upload the source code into our S3 bucket, now is the time for us to create the security group.

Creation of Security group:

The security group is important as it will help us to manage the traffic and which traffic is allowed into the instance and out from the instance. For example, if we want only this to host at port 80, and enable ICMP, this security group will help us to do this method. This acts as traffic and control which can and cannot flow into the instance. Firstly we have configured

the “Qaziwa private security group”, this is where this security group will allow SSH and database default port which is 3306. This is to ensure we are able to connect to the instance via SSH and able to fetch the database/tables from the web server to the database server.

The screenshot shows the AWS CloudFormation console with the following details:

- VPC Creation:** A new VPC named "Qaziwa VPC" is being created with the ID "vpc-090a8ad552bdc7ed6".
- Subnets:** Two public subnets ("Qaziwa Public Subnet 1" and "Qaziwa Public Subnet 2") and two private subnets ("Qaziwa Private Subnet 1" and "Qaziwa Private Subnet 2") are listed.
- Security Groups:**
 - Qaziwa private security group:** Security group ID "sg-0aba88ff4c2c181fc". It has an inbound rule for SSH (TCP port 22) and an outbound rule for MySQL (TCP port 3306).
 - Qaziwa public security group:** Security group ID "sg-0f6bd7a7a49643b7d". It has an inbound rule for SSH (TCP port 22) and an outbound rule for the internet.
- Outputs:** The VPC endpoint for Amazon RDS is listed as "Qaziwa RDS Endpoint".

We allow from any source for SSH as we want to connect from our laptop and also for MYSQL because we have not created the instance yet, so the specific IP address has not been assigned yet.

After creating the private security group, now is the time to create the public security group. This is where it is for the web server to access the internet. We have configured the public security group shown below.

The screenshot shows the AWS CloudFormation console with the following details:

- Public Security Group:** A new security group named "Qaziwa public security group" is being created with the ID "sg-0f6bd7a7a49643b7d".
- Inbound Rules:** An inbound rule for SSH (TCP port 22) is defined.
- Outbound Rules:** An outbound rule for the internet is defined.

Name	Security group ID	Security group name	VPC ID	Description	Owner
-	sg-0d1089b691aab8a9f	default	vpc-022432903ae41160e	default VPC security gr...	501
<input checked="" type="checkbox"/> Qaziwa public HTT...	sg-0f6bd7a7a49643b7d	Qaziwa public security ...	vpc-090a8ad552bcd7ed6	Qaziwa public security ...	501
-	sg-4d98734b	default	vpc-cb9a0bb6	default VPC security gr...	501

Inbound rules (3)					
<input type="button" value="C"/> Manage tags <input type="button" value="Edit inbound rules"/>					
	Type	Protocol	Port range	Source	Description
	SSH	TCP	22	0.0.0.0/0	-
	HTTPS	TCP	443	0.0.0.0/0	-
	All traffic	All	All	0.0.0.0/0	-

For the Public security group, we have enabled SSH for our configuration and port 443 which is HTTPS due to this instance server as a web server. We are able from any source because firstly we want to try access to the web service and to the SSH and also it provides availability to the users who access the website.

Since we know that due to the limited privileges that the AWS education has to offer and this cause we cannot use the NAT gateway. As such to improvise and mitigate this, we will be using an instance that will act as a NAT service.

Security Groups (1/5) Info					
<input type="button" value="C"/> Actions <input type="button" value="Create security group"/>					
<input type="button" value="Filter security groups"/>					
Name	Security group ID	Security group name	VPC ID	Description	Owner
-	sg-0d1089b691aab8a9f	default	vpc-022432903ae41160e	default VPC security gr...	501
<input checked="" type="checkbox"/> Qaziwa NAT security grp	sg-0ba7db7eb33d30e50	Qaziwa NAT security grp	vpc-090a8ad552bcd7ed6	launch-wizard-1 create...	501144450

Details					
Security group name <input type="button" value="Qaziwa NAT security grp"/>	Security group ID <input type="button" value="sg-0ba7db7eb33d30e50"/>	Description <input type="button" value="launch-wizard-1 created 2021-06-02T15:53:41.005+08:00"/>	VPC ID <input type="button" value="vpc-090a8ad552bcd7ed6"/>		
Owner <input type="button" value="501144450828"/>	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry			
Actions Create security group					

Security Groups (1/5) Info					
<input type="button" value="C"/> Actions <input type="button" value="Create security group"/>					
<input type="button" value="Filter security groups"/>					
Name	Security group ID	Security group name	VPC ID	Description	Owner
-	sg-0d1089b691aab8a9f	default	vpc-022432903ae41160e	default VPC security gr...	501
<input checked="" type="checkbox"/> Qaziwa NAT security grp	sg-0ba7db7eb33d30e50	Qaziwa NAT security grp	vpc-090a8ad552bcd7ed6	launch-wizard-1 create...	501144450

Details	Inbound rules	Outbound rules	Tags	
<input type="button" value="Edit inbound rules"/>				
Inbound rules (2)				
Type	Protocol	Port range	Source	Description - optional
All traffic	All	All	0.0.0.0/0	-
SSH	TCP	22	0.0.0.0/0	-

NAT security groups have enabled everything due to each instance having its own firewall or security rule and therefore, implementing this is much easier.

Creation of Elastic IP for web service:

Allocate an Elastic IP address for the web server. Without allocating, the IP addresses will keep changing after every refresh.

The screenshot shows the AWS Elastic IP Addresses console. At the top, there is a search bar labeled "Filter Elastic IP addresses". Below it is a table with columns: Name, Allocated IPv4 add..., Type, Allocation ID, Associated instance ID, and Private IP address. A single row is selected, showing "Web server elastic IP" as the name, "52.6.166.103" as the allocated IP, "Public IP" as the type, "eipalloc-0ca2a7acbe5b75259" as the allocation ID, "i-07e036f89a049c783" as the associated instance ID, and "10.0.1.84" as the private IP address. An orange "Allocate Elastic IP address" button is visible at the top right. Below the table is a "Summary" section with detailed information about the elastic IP, including its association with an instance and network interface.

We have created the Elastic IP in which we will be using later on for the instance implementation. This helps to provide a static public IP address in which it will not keep changing so that our DNS server will help to ensure that it will be able to locate the specific IP address. Now we have settled the configuration for the network, so next, we are ready to start recreating the instance and assign it to the load balancer and give auto-scaling, DNS, and much more to the instance so that it can provide the security and also scalability and even availability to the users.

Creation of web server instance:

To get started we created an Ubuntu instance with a micro size. The web server is then configured with a public IP to allow users to access the webserver. With the instance created we would ensure the port 22 is open for us to implement the website into the instance through there. With the SSH open, we can now connect to the instance via Putty through its public IP. Inside the command line, we would then run as root as cd into the /var/www/html folder to insert the code. Running the wget command we would insert the source code through the bucket URL of the Putty into the IP.

-> To upgrade the Ubuntu Machine and make sure it is the newest version:

```
ubuntu@ip-10-0-1-84:~$ sudo su
root@ip-10-0-1-84:/home/ubuntu# apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ub
```

Run apt-get update

Run apt-get upgrade

```
Reading package lists... Done
root@ip-10-0-1-84:/home/ubuntu# apt-get upgrade
Reading package lists... Done
```

-> Installing the Apache2 web server with its mods:

Run apt-get install apache2

```
Run apt-get install php libapache2-mod-php php-mysql
```

```
root@ip-10-0-1-84:/home/ubuntu# apt-get install php libapache2-mod-php php-mysql
Reading package lists... Done
```

->Getting the Website source code that we have put in the AWS S3 bucket:

```
Run wget https://qaziwabucket.s3.amazonaws.com/C300.zip
```

```
root@ip-10-0-1-84:/var/www/html# ls
root@ip-10-0-1-84:/var/www/html# wget https://qaziwabucket.s3.amazonaws.com/C300.zip
--2021-06-02 06:45:00-- https://qaziwabucket.s3.amazonaws.com/C300.zip
Resolving qaziwabucket.s3.amazonaws.com (qaziwabucket.s3.amazonaws.com) ... 52.216.128.
```

-> Installing unzips software to unzip a folder:

```
Run apt install unzip
```

-> Unzipping the C300.zip

```
Run unzip C300.zip
```

```
Run mv 'C300 (new)' C300 (To rename the folder)
```

```
      inflating: C300 (new) / PHPMailer/language/phpmailer.
root@ip-10-0-1-84:/var/www/html# ls
'C300 (new)'  C300.zip
root@ip-10-0-1-84:/var/www/html# mv 'C300 (new)' C300
root@ip-10-0-1-84:/var/www/html# ls
C300  C300.zip
```

-> Placing the source code into the /var/www/html

```
Run mv C300/* (Move all of the files and folders from the C300 folder out)
```

```
root@ip-10-0-1-84:/var/www/html# mv C300/* .
root@ip-10-0-1-84:/var/www/html# ls
AddBooks.php    GuestList.php        UserList.php      doAddBooks.php   doRegister.php
AdminList.php   Login.php          Verification.php doAddToCard.php  doVerification.php
C300           PHPMailer          WebDesign        doCartDelete.php include
C300.zip        PaymentPage.php   advertisement.js doDelete.php     index.php
Cart.php        Pictures          creatives       doEditBooks.php logoff.php
DeleteBooks.php Register.php      dbFunctions.php  doEditBooks2.php nbproject
EditBooks.php   SuccessfulPayment.php dbFunctions2.php doLogin.php    smtp
root@ip-10-0-1-84:/var/www/html#
```

-> Restart the apache2 service

```
Run service apache2 restart
```

```
root@ip-10-0-1-84:/var/www/html# service apache2 restart
root@ip-10-0-1-84:/var/www/html#
```

-> Change the source code to match with the database server IP address

```
Run nano /var/www/html/dbFunctions2.php
```

```
GNU nano 4.8                               dbFunctions2.php                                Modified
<?php

$db_host = "10.0.2.239";
$db_username = "qaziwa";
$db_password = "Password";
$db_name = "qaziwa";

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

```

Our summary command line we used for our Web Server installation to the last configuration of the dbfunctions2.php

```

1 install apt-get
2 apt-get update
3 apt-get upgrade
4 apt-get install apache2
5 apt-get install php libapache2-mod-php php-mcrypt php-mysql
6 apt-get install php libapache2-mod-php php-mcrypt php-mysql
7 apt-get install php libapache2-mod-php php-mysql
8 service apache2 restart
9 cd /var/www/html
10 ls
11 sudo nano
12 nano test.php
13 ls
14 rm index.html && test.php
15 ls
16 rm test.php
17 ls
18 wget https://qaziwabucket.s3.amazonaws.com/C300.zip
19 ls
20 unzip
21 apt install unzip
22 ls
23 unzip C300.zip
24 ls
25 mv 'C300 (new)' C300
26 ls
27 cd /var/www/html
28 ls
29 cd /var/www/html
30 ls
31 mv C300/* .
32 ls
33 service httpd start
34 service apache2 restart
35 ls

```

-> Change of the permission of the Pictures folder to enable adding images into the folder.

```

49  ls | grep "doA"
50  nano doAddBooks.php
51  ls
52  chmod 777 Pictures
53  ls -l
54  cd /var/www/html
55  ls
56  history
root@ip-10-0-1-84:/home/ubuntu#

```

We download the apache2 and the PHP so that we can run the code, then after that, we use wget to install the website source code in which to be used in the webserver. After installing the source code, we unzip it and store it in the directory /var/www/html and move the C300/ folder inside the HTML folder. After that, we restart the apache server and edit a source code tallied with the database server IP address.

Server Logs

- `/var/log/apache2/access.log`: By default, every request to your web server is recorded in this log file unless Apache is configured to do otherwise.
- `/var/log/apache2/error.log`: By default, all errors are recorded in this file. The `LogLevel` directive in the Apache configuration specifies how much detail the error logs will contain.

Now from above, here is the logs directory we can check and configuration of the apache server. Once we have done setting up the web server, now is the time for us to implement the database server to fully make use of the website. But before the database is being set up we need to create the NAT instance so that we are able to install the phpmyadmin for our database.

Creation of NAT instance:

Since AWS education, we cannot use the NAT function, therefore, we need to use the configuration of the NAT instance instead. By doing this it enables the private IP address to connect to the internet but the internet cannot connect to the private IP, this function helps to keep the database server secure and at the same time enables it to download the resources too. Below shows the configuration of the NAT instance

First, we create an instance by using a NAT instance.

-> Creation of the NAT instance

Goto EC2 > Launch Instance > Community AMIs > search NAT > click the amzn-ami-vpc-nat-hnm-2018.03.0.20181116-x86_64-ebs

Step 1: Choose an Amazon Machine Image (AMI)
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Goto Configuration Instance Details → Select our own VPC (Qaziwa VPC) → Select the public Subnet) > Click next: Add Storage

Step 3: Configure Instance Details
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 Launch into Auto Scaling Group:

Purchasing option: Request Spot Instances

Network: `vpc-090a8ad562bd07ed6 | Qaziwa VPC`
Subnet: `subnet-0bca6218d2bed2e5 | Qaziwa public subnet`
248 IP Addresses available

Auto-assign Public IP:

Placement group: Add instance to placement group

Capacity Reservation:

Domain join directory:

IAM role:

Shutdown behavior:

Stop - Hibernate behavior: Enable hibernation as an additional stop behavior

Cancel Previous Review and Launch Next: Add Storage

Goto to Add Tags → Key in the name and value (shown below) > Click Next: Configure Security Group

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.

A copy of a tag can be applied to volumes, instances or both.

Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	(128 characters maximum)	Value	(256 characters maximum)	Instances	Volumes	Network Interfaces
Name		Qaziwa NAT instance		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Add another tag	(Up to 50 tags maximum)					

Goto Configure Security Group → Click the security group that we have already created for NAT instances.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group
 Select an existing security group

Security Group ID	Name	Description	Actions
sg-03c65391d4e841f9a	Database Test Server	launch-wizard-1 created 2021-06-18T11:31:02.982+08:00	Copy to new
sg-0b1246adfb0b167c3c	default	default VPC security group	Copy to new
sg-0373078b558da33a2d	launch-wizard-1	launch-wizard-1 created 2021-06-18T12:16:53.142+08:00	Copy to new
sg-0ba7db7eb33d30e50	Qaziwa NAT security grp	launch-wizard-1 created 2021-06-02T15:53:41.005+08:00	Copy to new
sg-0ba8a8ff4c2c181fc	Qaziwa private security grp	Qaziwa private security grp	Copy to new
sg-0f6bd7a7a49643b7d	Qaziwa public security grp	Qaziwa public security grp	Copy to new

Inbound rules for sg-0ba7db7eb33d30e50 (Selected security groups: sg-0ba7db7eb33d30e50)

Type	Protocol	Port Range	Source	Description
All traffic	All	All	0.0.0.0/0	
SSH	TCP	22	0.0.0.0/0	

[Cancel](#) [Previous](#) [Review and Launch](#)

We want to install the database server inside the private group, therefore, we have enabled all traffic. After that click review and launch of this NAT instance

Here is the summary of the configuration of our NAT instance:

Instance ID	Public IPv4 address	Private IPv4 addresses
i-069952ee56c25479e (Qaziwa NAT instance)	18.209.22.28 open address	10.0.1.105
Instance state	Public IPv4 DNS	Private IP4 DNS
Running	ec-18-209-22-28.compute-1.amazonaws.com open address	ip-10-0-1-105.ec2.internal
Instance type	Elastic IP addresses	VPC ID
t2.micro	-	vpc-090a8ad552bdc7ed6 (QaziWa VPC)
AWS Compute Optimizer finding	IAM Role	Subnet ID
>User: arn:aws:sts::501144450828:assumed-role/vocstartsoft/user1406304=19031563@myrp.edu.sg is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: * with an explicit deny	-	subnet-0bccca6218d2bed2e5 (Qaziwa public subnet)
Retry		
Instance details		

After creating an instance, we need to stop the source/destination checking and click on save

EC2 > Instances > i-069952ee56c25479e > Change source / destination check

Source / destination check [Info](#)

Each EC2 instance performs source and destination checks by default. The instance must be the source or destination of all the traffic it sends and receives.

Instance ID
 [i-069952ee56c25479e](#) (Qaziwa NAT instance)

Network interface [Info](#)
 [eni-075a491ffe78f3ca4](#) (Qaziwa NAT instance)

Source / destination checking [Info](#)
 Stop

If this is a NAT instance, you must stop source / destination checking. A NAT instance must be able to send and receive traffic when the source or destination is not itself.

AWS CLI Command

```
aws ec2 modify-instance-attribute --instance-id=i-069952ee56c25479e --no-source-dest-check
```

After changing the setting, then we need to link it back to the route table.

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC
<input checked="" type="checkbox"/> Qaziwa Route Table for NAT	rtb-0856a597fe2623a4d	subnet-04e9d1ba0d738e7d6 / Qaziwa private subnet	-	No	vpc-090a8ad552bcd7ed6 Qaziwa VPC
<input type="checkbox"/> QaZiWa Route Table for Internet Gateway	rtb-073c5b0a953c1945f	subnet-0bccaa6218d2bed2e5 / Qaziwa public subnet	-	Yes	vpc-090a8ad552bcd7ed6 Qaziwa VPC
	rtb-0840cc76		-	Var	vpc-090a8ad552bcd7ed6 Qaziwa VPC

rtb-0856a597fe2623a4d / Qaziwa Route Table for NAT

[Details](#) [Routes](#) [Subnet associations](#) [Edge associations](#) [Route propagation](#) [Tags](#)

Details

Route table ID <input type="checkbox"/> rtb-0856a597fe2623a4d	Main <input type="checkbox"/> No	Explicit subnet associations <input type="checkbox"/> subnet-04e9d1ba0d738e7d6 / Qaziwa private subnet	Edge associations
VPC <input type="checkbox"/> vpc-090a8ad552bcd7ed6 Qaziwa VPC	Owner ID <input type="checkbox"/> 501144450828		-

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC
<input checked="" type="checkbox"/> Qaziwa Route Table for NAT	rtb-0856a597fe2623a4d	subnet-04e9d1ba0d738e7d6 / Qaziwa private subnet	-	No	vpc-090a8ad552bcd7ed6 Qaziwa VPC
<input type="checkbox"/> QaZiWa Route Table for Internet Gateway	rtb-073c5b0a953c1945f	subnet-0bccaa6218d2bed2e5 / Qaziwa public subnet	-	Yes	vpc-090a8ad552bcd7ed6 Qaziwa VPC
	rtb-0840cc76		-	Var	vpc-090a8ad552bcd7ed6 Qaziwa VPC

rtb-0856a597fe2623a4d / Qaziwa Route Table for NAT

[Details](#) [Routes](#) [Subnet associations](#) [Edge associations](#) [Route propagation](#) [Tags](#)

Routes (2)

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	eni-075a491ffe78f3ca4	Active	No

After changing the Route Table, we also need to change the private subnet so that the instance from the private subnet can connect to the NAT instance, and from the NAT instance the Traffic and flow out to the routing table and to the internet gateway and out to the internet. Since the NAT instance is fully operational now is the time to set up the database instance.

Creation of database server:

```
Last login: Wed Jun  2 10:42:59 2021 from 119.56.96.205
ubuntu@ip-10-0-1-84:~$ history
 1 ping 10.0.2.239
 2 ping google.com
 3 sudo su
 4 cd /var/www/html
 5 ls
 6 sudo su
 7 ls
 8 sudo ssh ubuntu@10.0.2.239 -i QaziwaKey.pem
 9 history
ubuntu@ip-10-0-1-84:~$
```

From the Web Server, we tried to ping into the database server to check the connection with it. After getting the response, we tried to copy the SSH private key and paste it into the webserver using sudo nano QaziwaKey.pem. Then we change the permission to 0400 and use ssh ubuntu@10.0.2.239 -i QaziwaKey.pem to get into the database.

```
8 updates can be applied immediately.
7 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

*** System restart required ***
Last login: Wed Jun  2 09:36:24 2021 from 10.0.1.84
ubuntu@ip-10-0-2-239:~$
```

After getting into the Database, we tried to ping google.com and to see if there was any response. In addition, we have already set up a NAT instance that enables us to connect to the internet without using any public IP address from the Database Server instance.

After we can see that it can connect into the database, we then use the command “apt-get install lamp-server^” → use “apt-get install PHPMyAdmin -y” (choose apache2 > yes > key in password > confirm password) after downloading the phpmyadmin, we then enter into the mysql database and create a user using “mysql -p -u root” → “CREATE USER ‘qaziwa’@’%’ IDENTIFIED BY ‘<The password>’;” → “GRANT ALL PRIVILEGES ON *.* TO ‘qaziwa’@’%’ WITH GRANT OPTION;”

```
mysql> CREATE USER 'qaziwa'@'%' IDENTIFIED BY 'Password';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'qaziwa'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.01 sec)
```

After creating the user, then I need to go to the config.inc.php to edit the file and make out my own profile inside it. Shown below:

```

GNU nano 4.8
/etc/phpmyadmin/config.inc.php

// $cfg['Servers'][$i]['tracking'] = 'pma_tracking';
// $cfg['Servers'][$i]['userconfig'] = 'pma_userconfig';
// $cfg['Servers'][$i]['recent'] = 'pma_recent';
// $cfg['Servers'][$i]['favorite'] = 'pma_favorite';
// $cfg['Servers'][$i]['users'] = 'pma_users';
// $cfg['Servers'][$i]['usergroups'] = 'pma_usergroups';
// $cfg['Servers'][$i]['navigationhiding'] = 'pma_navigationhiding';
// $cfg['Servers'][$i]['savedsearches'] = 'pma_savedsearches';
// $cfg['Servers'][$i]['central_columns'] = 'pma_central_columns';
// $cfg['Servers'][$i]['export_templates'] = 'pma_export_templates';
/* Contrib / Swekey authentication */
// $cfg['Servers'][$i]['auth_swekey_config'] = '/etc/swekey-pma.conf';

/*
 * End of servers configuration
 */

/*
 * Directories for saving/loading files from server
 */
$cfo['UploadDir'] = '';
$cfo['SaveDir'] = '';

/* Support additional configurations */
foreach (glob('/etc/phpmyadmin/conf.d/*.php') as $filename)
{
    include($filename);
}

$i++;
$cfo['Servers'][$i]['host'] = '10.0.2.239';
$cfo['Servers'][$i]['port'] = '3306';
$cfo['Servers'][$i]['connect_type'] = 'tcp';
$cfo['Servers'][$i]['extension'] = 'mysql';
$cfo['Servers'][$i]['auth_type'] = 'config';
$cfo['Servers'][$i]['user'] = 'gazawa';
$cfo['Servers'][$i]['password'] = 'Password';

```

After configuring we need to go to the nano /etc/mysql/mysql.conf.d/mysqld.cnf and comment out the bind-address. Shown below:

```

# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
user          = mysql
# pid-file     = /var/run/mysqld/mysqld.pid
# socket       = /var/run/mysqld/mysqld.sock
# port         = 3306
# datadir      = /var/lib/mysql

#
# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html
# tmpdir        = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
# bind-address    = 127.0.0.1
# mysqlx-bind-address = 127.0.0.1
#
# * Fine Tuning
#
key_buffer_size      = 16M
# max_allowed_packet = 64M
# thread_stack        = 256K

# thread_cache_size   = -1

```

[Read 78 lines]

After save the file, and restart the mysql using “systemctl restart mysql”

After that we connect to the user using “mysql -h 10.0.2.239 -u qaziwa -p” > then now we want to create a database using “CREATE DATABASE qaziwa;” then click exit then import the qaziwa.sql from the S3 bucket using wget and the URL. Shown below

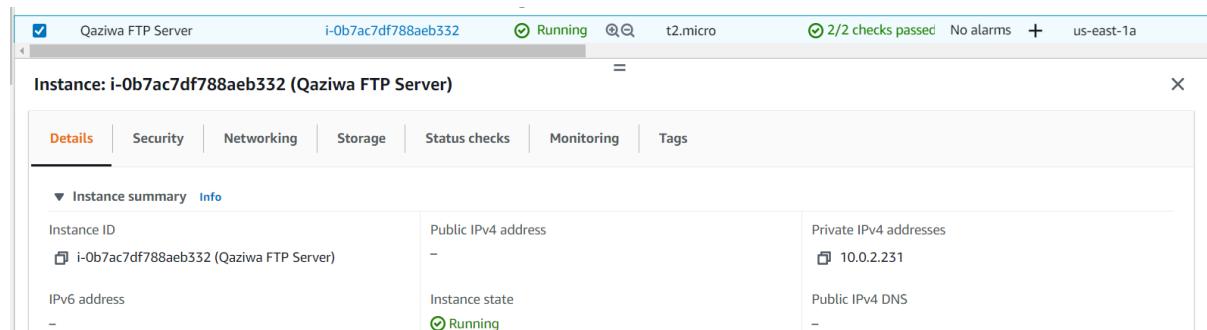
```
8 wget https://qaziwabucket.s3.amazonaws.com/qaziwa.sql
9 ls
10 mysql -h 10.0.2.239 -u qaziwa -p qaziwa < qaziwa.sql
11 sudo su
12 history
ubuntu@ip-10-0-2-239:~$ █
```

After importing the qaziwa.sql then we will import it to the database using the command “mysql -h 10.0.2.239 -u qaziwa -p qaziwa < qaziwa.sql”. Then from there, we test the database and the website.

Since we have fully configured the web server, NAT instance and database server now are the time to create the FTP Server instance.

Creation of FTP server instance:

FTP server helps us to store and get any files content be it an image file, text file, or any other files. Since Load Balancing is needed, we have an idea that if we upload an image file to one website, the other website may not be updated. This can cause users to not have a full experience when using the website and this could lead to decrease of its functions. As such we plan and implement a FTP server in which it will help us to get and store image files whenever the web server uploads or needs the image to display to the user. Since our website has an upload function, we need to upload it into our FTP server in which when the web server needs the file, it will connect to the FTP server and take in the file that is needed which it will be depending on the database server image file name. As such now is the configuration of the FTP server. First we create an instance and ensure that the FTP server is in the private network and the same network as the database instance.



After creation of the instance, Putty inside the web server and then ssh inside the FTP server. Since we already created a NAT instance, we can use it to down a FTP resource package.

To install the FTP server we need be in root account and then run apt-get update and apt-get install vsftpd

```
sudo apt-get update
sudo apt-get install vsftpd
```

After the installation has been completed, the service is disable by default, so therefore, we need to start it automatically.

```
systemctl start vsftpd  
systemctl enable vsftpd
```

Since we have already install, now is to configure the vsftpd.conf which is located at /etc/vsftpd.conf

```
sudo nano /etc/vsftpd.conf
```

After this we enable the requirements shown below

```
anonymous_enable=NO          # disable anonymous login  
local_enable=YES            # permit local logins  
write_enable=YES             # enable FTP commands which change the  
                           # value of umask for file creation for  
                           # users  
local_umask=022              # enable showing of messages when user  
                           # a log file will be maintained detailing  
                           # use port 20 (ftp-data) on the server  
dirmessage_enable=YES        # keep standard log file format  
xferlog_enable=YES           # prevent vsftpd from running in stand-alone  
                           # mode  
connect_from_port_20=YES     # vsftpd will listen on an IPv6 socket  
                           # name of the PAM service vsftpd will  
                           # enable vsftpd to load a list of user  
                           # turn on tcp wrappers  
  
listen=NO  
listen_ipv6=YES  
pam_service_name=vsftpd  
userlist_enable=YES  
tcp_wrappers=YES  
  
chroot_local_user=YES  
allow_writeable_chroot=YES  
  
userlist_enable=YES  
userlist_file=/etc/vsftpd.userlist  
userlist_deny=NO
```

After enabling the setting shown above, we then need to restart the FTP service using the systemctl restart vsftpd.

Once it has been restarted, we need to create the username called Qaziwa with a password as a Password. So, we use the command “useradd -m -c “Qaziwa” -s /bin/bash Qaziwa” and then we need to set the password so we need to key in “passwd Qaziwa” and from there they will ask for the key in the password. The password for this will be Password.

After this, we want to add user to the vsftpd.userlist by using “echo “Qaziwa” | sudo tee -a /etc/vsftpd.userlist” and to double clarify if the Qaziwa username is being added we do a “cat /etc/vsftpd.userlist”.

Now is the time to test the FTP service. Firstly we try to login into an Anonymous account to see if we can connect or not. Since we know that the anonymous setting in the FTP server has been set to no, which means it would not let us login as anonymous.

```
ubuntu@ip-10-0-1-48:~$ ftp 10.0.2.231
Connected to 10.0.2.231.
220 (vsFTPd 3.0.3)
Name (10.0.2.231:ubuntu): Anonymous
530 Permission denied.
Login failed.
ftp> exit
221 Goodbye.
ubuntu@ip-10-0-1-48:~$ ftp 10.0.2.231
Connected to 10.0.2.231.
220 (vsFTPd 3.0.3)
Name (10.0.2.231:ubuntu): anonymous
530 Permission denied.
Login failed.
ftp> █
```

As shown above, we can see that we are unable to run as an anonymous user.

So now let use with the user we have created in the previous section with the username “Qaziwa” and password “Password”

```
ubuntu@ip-10-0-1-48:~$ ftp 10.0.2.231
Connected to 10.0.2.231.
220 (vsFTPd 3.0.3)
Name (10.0.2.231:ubuntu): Qaziwa
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxrwx---    2 1001      1001        4096 Jul 29 18:14 files
226 Directory send OK.
ftp> █
```

We have successfully logged in and are able to log as a Qaziwa user in which will be used to store all of the image files that the admin has uploaded.

Now is to configure the home directories for the FTP user.

Firstly we do:

```
mkdir /home/Qaziwa/ftp/files
```

```
chown -R Qaziwa:Qaziwa /home/Qaziwa/ftp/files  
chmod -R 0770 /home/Qaziwa/ftp/files
```

After that, we would add the settings shown below

```
user_sub_token=$USER          # inserts the username in the local root  
local_root=/home/$USER/ftp    # defines any users local root directory
```

After that do a restart of the FTP service by using the command “systemctl restart vsftpd”

Since we have restarted, now we test if we have a file folder and are able to access the file folder.

```
ubuntu@ip-10-0-1-48:~$ ftp 10.0.2.231  
Connected to 10.0.2.231.  
220 (vsFTPd 3.0.3)  
Name (10.0.2.231:ubuntu): Qaziwa  
331 Please specify the password.  
Password:  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp> ls  
200 PORT command successful. Consider using PASV.  
150 Here comes the directory listing.  
drwxrwx--- 2 1001 1001 4096 Jul 29 18:14 files  
226 Directory send OK.  
ftp> cd files  
250 Directory successfully changed.  
ftp> ls  
200 PORT command successful. Consider using PASV.  
150 Here comes the directory listing.  
-rw-r--r-- 1 1001 1001 167908 Jul 28 20:30 beano_front.jpg  
-rw-r--r-- 1 1001 1001 92954 Jul 28 20:34 cat_hat_front.jpg  
-rw-r--r-- 1 1001 1001 2034 Jul 29 16:35 download.jpg  
-rw-r--r-- 1 1001 1001 105143 Jul 28 20:31 garfield_front.jpg  
-rw-r--r-- 1 1001 1001 95222 Jul 28 20:31 ghost_stories_front.jpg  
-rw-r--r-- 1 1001 1001 110324 Jul 28 20:32 hard_luck_front.jpg  
-rw-r--r-- 1 1001 1001 5407 Jul 29 18:14 hello.jpg  
-rw-r--r-- 1 1001 1001 100374 Jul 28 20:31 insomnia_front.jpg  
-rw-r--r-- 1 1001 1001 140113 Jul 28 20:31 kid_front.jpg  
-rw-r--r-- 1 1001 1001 108025 Jul 28 20:33 long_haul_front.jpg  
-rw-r--r-- 1 1001 1001 122984 Jul 28 20:31 old_school_front.jpg  
-rw-r--r-- 1 1001 1001 99563 Jul 28 20:32 passing_front.jpg  
-rw-r--r-- 1 1001 1001 174397 Jul 28 20:32 transformers_front.jpg  
226 Directory send OK.  
ftp> █
```

Now we are able to get and put the image file inside the FTP server. Once that is done then we need to upload the file and download the files from the FTP server using the PHP file.

After intensive research, we manage to upload and download the file to and from the FTP server. This can be seen below.

Below is the upload from the web server to the FTP server. This is when the admin wants to upload a new book image so that the user can see the book image.

```

$ftp_host = "10.0.2.231";
$ftp_username = "Qaziwa";
$ftp_password = "Password";

// open an FTP connection
$conn_id = ftp_connect($ftp_host) or die("Couldn't connect to $ftp_host");

// login to FTP server
$ftp_login = ftp_login($conn_id, $ftp_username, $ftp_password);

// local & server file path
$localFilePath = "/var/www/html/Pictures/$fileName";
$remoteFilePath = "files/$fileName";

// try to upload file
if (ftp_put($conn_id, $remoteFilePath, $localFilePath, FTP_BINARY)) {
    echo "File transfer successful - $localFilePath";
} else {
    echo "There was an error while uploading $localFilePath";
}

// close the connection
ftp_close($conn_id);

```

Next is to download the file from the FTP server to the web server. This is much harder as we do not want all images to be downloaded to our web server as this can cause slowness. As such we have come up with an idea to only accept the image name that tally with the database name.

	<input type="checkbox"/> Ed	<input type="checkbox"/> title	<input type="checkbox"/> id	<input type="checkbox"/> title	<input type="checkbox"/> author	<input type="checkbox"/> summary	<input type="checkbox"/> rating	<input type="checkbox"/> price	<input type="checkbox"/> quantity	<input type="checkbox"/> picture_front
Click the drop-down arrow to toggle column's visibility.										
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	1	The Cat In The Hat	Dr. Suess	Poor Dick and Sally. It's cold and wet and they're...	5	13.57	8
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	3	Garfield 30 Years	Jim Davis	30 years of laugh and lasagna bundled into one boo...	4	8.16	43
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	4	True Singapore Ghost Stories 11	Russel Lee	Russel Lee investigates witchcraft and uncovers it...	2	10.59	7
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	5	Diary of a Wimpy Kid Hard Luck	Jeff Kinney	Greg Heffley's on a losing streak. His best friend...	4	14.93	0
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	6	Insomnia	Stephen King	You'll lose a lot of sleep. Ralph does. At first h...	3	16.95	14
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	7	When I was a kid 2	Boey	A continuation of the hit book "When I Was a Kid" ...	3	19.24	4
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	8	Diary of a Wimpy Kid Long Haul	Jeff Kinney	A family road trip is supposed to be a lot of fun ...	4	20.87	20
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	9	Diary of a Wimpy Kid Old School	Jeff Kinney	Life was better in the old days. Or was it? That's...	2	20.87	13
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	10	As I was Passing II	Adibah Amin	Adibah Amin, the celebrated chronicler of everyday...	3	19.90	3
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	11	Transformers Saga of the Allspark	Simon Furman	Saga of the Allspark reveals untold events from th...	5	17.99	25

Our database consists of the image file name in which we can download the specific books we have implemented. If the admin adds a new book, the database will be uploaded, and from the web server, which will tally the amount of file name to be downloaded and get from the FTP server.

```

$ftp_host = "10.0.2.231";
$ftp_username = "Qaziwa";
$ftp_password = "Password";

// open an FTP connection
$conn_id = ftp_connect($ftp_host) or die("Couldn't connect to $ftp_host");

// login to FTP server
$ftp_login = ftp_login($conn_id, $ftp_username, $ftp_password);

try {
    $conn99 = new PDO("mysql:host=$db_host;dbname=$db_name", $db_username, $db_password);
    $conn99->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $stmt99 = $conn99->prepare("SELECT * FROM books");

    $stmt99->execute();
} catch (Exception $ex) {
    echo "Error: " . $ex->getMessage();
}
$conn99 = null;

// $query2 = "SELECT * FROM books";
// To secure, use mysqli_escape_string or prepared statements
// $result2 = mysqli_query($link, $query2) or die(mysqli_error($link));

while ($row99 = $stmt99->fetch()) {
    $arrProducts99[] = $row99;
}

for ($i = 0; $i < count($arrProducts99); $i++) {
    // $title = $arrProducts[$i]['title'];
    // $author = $arrProducts[$i]['author'];
    // $summary = $arrProducts[$i]['summary'];
    // $rating = $arrProducts[$i]['rating'];
    // $price = $arrProducts[$i]['price'];
    // $quantity = $arrProducts[$i]['quantity'];
    $picture99 = $arrProducts99[$i]['picture_front'];
    $localFilePath = "/var/www/html/Pictures/$picture99";
    $remoteFilePath = "files/$picture99";

    // try to upload file
    if (ftp_get($conn_id, $localFilePath, $remoteFilePath, FTP_BINARY)) {
        echo "File transfer successful - $localFilePath";
    } else {
        echo "There was an error while uploading $localFilePath";
    }
}
ftp_close($conn_id);

```

As shown above, first we connect to the FTP server and then we connect to the database server by getting in the file name that is inside the database server. After it receives all of the file names, then it will send over to the web server and from the web server, it will send to the FTP server and ask for the specific file name that is needed to be downloaded. Upon successful download, a message will appear stating a “File transfer successful” and then the file that has been downloaded. As such we can use this FTP server as our Upload and download of files. In addition, we also have uploaded the file in our web server folder in which it could act as a backup if there is a need to use it. After all of this, now is the time to create the snapshot for our instance.

Creation of Instance Snapshot:

First, we go into our targeted instance and scroll to its root device and click on the EBS ID to be redirected to the volume interface.

The screenshot shows the 'Block devices' section of the AWS Management Console. A table lists a single volume:

Volume ID	Device name	Volume size (GiB)	Attachment status
vol-09b211f8efe00ca7d	/dev/sda1	8	Attached

We would then go to actions and click on create a snapshot to snapshot the instance volume.

The screenshot shows the 'Actions' dropdown menu for a selected volume. The 'Create Snapshot' option is highlighted in orange.

- Modify Volume
- Create Snapshot
- Create Snapshot Lifecycle Policy
- Delete Volume
- Attach Volume
- Detach Volume
- Force Detach Volume
- Change Auto-Enable IO Setting
- Add/Edit Tags

We would then label the snapshot we are creating and in this case, since we are snapshotting the web server we would label it as such.

The screenshot shows the 'Create Snapshot' dialog box. It includes fields for Volume, Description, and Encryption, along with a tagging section and a 'Create Snapshot' button.

Volumes > Create Snapshot

Create Snapshot

Volume vol-09b211f8efe00ca7d ⓘ

Description Snapshot for Web Server 1 ⓘ

Encrypted Not Encrypted ⓘ

Key (128 characters maximum) Value (256 characters maximum)

Name Web Server Qaziwa 1 Snapshot ⌂

Add Tag 49 remaining (Up to 50 tags maximum)

* Required Cancel Create Snapshot

After snapshotting all the necessary instances we can go into the snapshot tab and locate all the snapshot that has been created.

Snapshot ID	Snapshot Name	Size	Created By
snap-0557dea4013...	Database Vo...	8 GiB	Created on 2/6/2021
snap-0d758b25fa0c...	Web Server ...	8 GiB	Snapshot on 16/6/2021
snap-0f9dceeb7747...	NAT Instanc...	8 GiB	Snapshot on 2/6/2021

We have successfully created the instance snapshot, so all of the information will be safe inside and if there is any instance accidentally terminated, we still can use this snapshot to recover all of our saved work. Now is the time to show how we can recover the instance configuration back to the original after an instance has been accidentally terminated.

Using Instance Snapshot:

First, we would need to take note of the Root Device of the instance that we are taking the snapshot from in this case it would be /dev/sda1

Instance: i-0eec8b9304eb2e77e (Web Server Qaziwa 1)

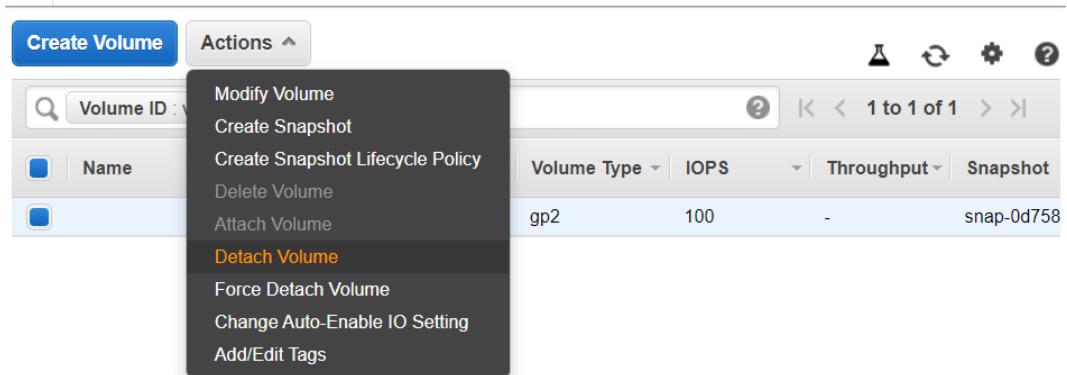
Root device details	
Root device name	Root device type
/dev/sda1	EBS

Next, we would create a new instance with all the same settings as the instance that we are taking the snapshot from ensuring that it is in the same availability zone and all that.

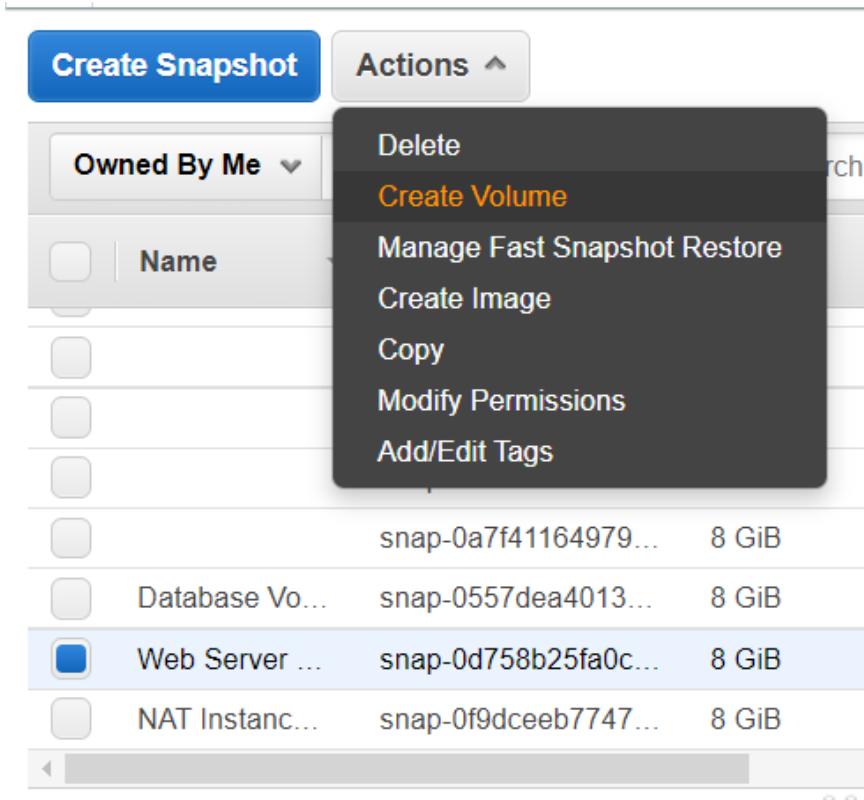
Instance: i-015628497df6338a4 (Web Server Qaziwa 2)

Details	Security	Networking	Storage	Status checks	Monitoring
Tags					
Instance summary <small>Info</small>					
Instance ID		Public IPv4 address			
i-015628497df6338a4 (Web Server Qaziwa)		-			

With the instance created we can now go to the volume and detach the current volume (ensure instance is turned off)



Once detach going back to the snapshot tab we would then choose the snapshot we are taking and create a new volume from actions.



Make sure the volume is located in the same availability zone as the instance.

Create Volume

Snapshot ID snap-0d758b25fa0cff804 (Web Server Volume Snapshot)

Volume Type General Purpose SSD (gp2) 

Size (GiB) 8

(Min: 1 GiB, Max: 16384 GiB) 

IOPS 100 / 3000

(Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS)



Throughput (MB/s) Not applicable 

Availability Zone* us-east-1a 

Fast Snapshot Restore Not enabled 

Encryption Encrypt this volume

After creating the volume, we would take note of the volume ID

[Snapshots](#) > Create Volume

Create Volume

 Create Volume Request Succeeded

Volume Id vol-053e35e46afedf2e9

Close

Going into the volumes tab we would locate our newly created volume that has the instance snapshot and attach it to our instance.

	Volume Type	IOPS	Throughput	Snapshot
gp2	100	-	snap-0d758	
gp2	100	-	snap-093cf8	
gp2	100	-	snap-093cf8	
gp2	100	-	snap-0447d	

We would then need to specify the instance we would be attaching it to, in this case, it would be Web Server 2 and as such we would key in the instance ID of that instance or just search it up by name. We would also need to ensure the device is the same as the root device that was mentioned earlier for successful restoration.

Attach Volume

Volume ⓘ
vol-053e35e46afedf2e9 (test volume) in us-east-1a

Instance ⓘ
i-0eec8b9304eb2e77e in us-east-1a

Device ⓘ
/dev/sda1
Linux Devices: /dev/sdf through /dev/sdp

Note: Newer Linux kernels may rename your devices to /dev/xvdf through /dev/xvdp internally, even when the device name entered here (and shown in the details) is /dev/sdf through /dev/sdp.

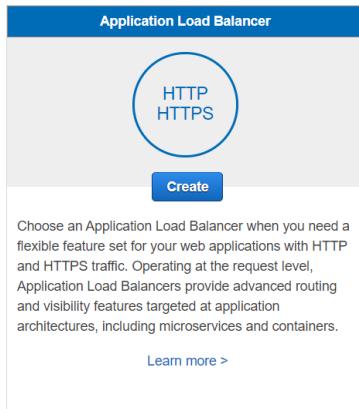
Cancel **Attach**

If there is any accidental termination of the instance, we can recover it using the snapshot that we have done. Since of the instance settings and configuration we have done, now is the time to create the load balancer in which to provide distribution of load evenly to the web server.

Creating of Load balancing:

Since our server may experience heavy network traffic, a load balancer will be implemented for disturbing incoming network traffic across multiple servers/targets to prevent instances from crashing in multiple availability zones. Load balancing also monitors the health of registered targets ensuring it routes traffic to only healthy targets. The few components in a load balancer are the load balancer itself, listener and target group. The load balancer will distribute traffic as mentioned earlier. The listener will check for connection requests from the client from our configuration. For every target group, it will route traffic to registered targets which is the instance. Other than that, the load balancer needs to support at least 2 availability zones, so the public subnet 2 & 2 we have created above, we can utilize it and use the load balancer to support the two public subnets with 2 different availability zones.

In our network, we use an application load balancer (ALB) which allows configuring and routing incoming end-user traffic to applications based on the public cloud. ALB inspect packers and will create access points to HTTP and HTTPS header. The ALB in our network is connected from our instances. In addition, we also configure stickiness in our ALB so that it only sticks to 1 instance at a time.



We choose ALB as our load balancer

Then go to configure load balancer → Name (QaizwaLoadBalancer) → internet-facing → IPV4 → HTTP (port 80) → Qaziwa VPC → Qaziwa Public Subnet 1 & 2 > then go to configure security settings

Step 1: Configure Load Balancer

Basic Configuration

To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network. The default configuration is an Internet-facing load balancer in the selected network with a listener that receives HTTP traffic on port 80.

Name: QaizwaLoadBalancer

Scheme: internet-facing internal

IP address type: IPv4

Listeners

A listener is a process that checks for connection requests, using the protocol and port that you configured.

Load Balancer Protocol	Load Balancer Port
HTTP	80

Add Listener

Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only one subnet per Availability Zone. You must specify subnets from at least two Availability Zones to increase the availability of your load balancer.

VPC	Subnet	IP Address	Assigned by
vpc-099aa8af52dc7e6f (10.0.0.16) Qaziwa VPC	us-east-1a	subnet-dccac119cdedc5 (Qaziwa public subnet)	Assigned by AWS
	us-east-1b	subnet-0563aaac151cafb8 (Qaziwa public subnet)	Assigned by AWS

Add-on services

Additional AWS services can be integrated with this load balancer at launch when you enable them below. You can also add these and other services after your load balancer is created by reviewing the "Integrated Services" tab for the selected load balancer.

AWS Global Accelerator Create an accelerator to get static IP addresses and improve the performance and availability of your

The configure security settings leave as default due to we did not enable HTTPS (due to we do not have the AWS privilege to do that) > Click Next: Configure Security Group → We choose Qaziwa Public Security Group (It enable port 80 which is HTTP)

1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

Step 3: Configure Security Groups

A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new security group or select an existing one.

Assign a security group Create a new security group Select an existing security group

Filter [VPC security groups ▾]

Security Group ID	Name	Description	Actions
sg-03c65391d4e841f9a	Database Test Server	launch-wizard-1 created 2021-06-18T11:31:02.982+08:00	Copy to new
sg-0b1246adf0b167c3c	default	default VPC security group	Copy to new
sg-0373078b58da83a2d	launch-wizard-1	launch-wizard-1 created 2021-06-18T12:16:53.142+08:00	Copy to new
sg-0ba7db7eb3d30e50	Qaziwa NAT security grp	launch-wizard-1 created 2021-06-02T15:53:41.005+08:00	Copy to new
sg-0aba88ff4c2c181fc	Qaziwa private security grp	Qaziwa private security grp	Copy to new
sg-0f6bd7a7a49643b7d	Qaziwa public security grp	Qaziwa public security grp	Copy to new

After this, create a new target group in which the instance will be inside this target group.

So, go to Configure Routing → Target group (new target group) → Name (QaziwaTargetGroup) → Target type (Instance) → Protocol (HTTP, port 80) then click next: register targets.

1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

Step 4: Configure Routing

Your load balancer routes requests to the targets in this target group using the protocol and port that you specify here. It also performs health checks on the targets using these settings. The target group you specify in this step will apply to all of the listeners configured on this load balancer. You can edit or add listeners after the load balancer is created.

Target group

Target group

Name

Target type Instance IP Lambda function

Protocol

Port

Protocol version HTTP1
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.
 HTTP2
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.
 gRPC
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Health checks

Protocol

Path

[Advanced health check settings](#)

[Cancel](#) [Previous](#) [Next: Register Targets](#)

After this choose an instance, then add to registered on port 80 then next: review and then create

1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

Step 5: Register Targets

Register targets with your target group. If you register a target in an enabled Availability Zone, the load balancer starts routing requests to the targets as soon as the registration process completes and the target passes the initial health checks.

Registered targets

To deregister instances, select one or more registered instances and then click Remove.

<input type="checkbox"/>	Instance	Name	Port	State	Security groups	Zone
<input type="checkbox"/>	i-015628497df6338a4	Web Server Qaziwa 2	80	running	Qaziwa public security grp	us-east-1b
<input type="checkbox"/>	i-0eec8b9304eb2e77e	Web Server Qaziwa 1	80	running	Qaziwa public security grp	us-east-1a

Instances

To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

Add to registered on port 80

<input type="checkbox"/>	Instance	Name	State	Security	Zone	Subnet ID	Subnet CIDR
<input checked="" type="checkbox"/>	i-015628497df6...	Web Server Qa...	running	Qaziwa public s...	us-east-1b	subnet-0d563aaca151ca8f8	10.0.3.0/24
<input type="checkbox"/>	i-0ed2ea0768b...	Database Serv...	running	Qaziwa private ...	us-east-1a	subnet-04e9d1ba0d738e7d6	10.0.2.0/24
<input checked="" type="checkbox"/>	i-0eec8b9304e...	Web Server Qa...	running	Qaziwa public s...	us-east-1a	subnet-0bccaa6218d2bed2e5	10.0.1.0/24

EC2 > Target groups

Target groups (1/1) [info](#)

[Create target group](#)

<input checked="" type="checkbox"/>	Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
<input checked="" type="checkbox"/>	QaziwaTargetGroup	arn:aws:elasticloadbalancin...	80	HTTP	Instance	QaziwaLoadBalancer	vpc-090aa8ad552hdc7edf

Registered targets (2)

[Edit](#)

<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status details
<input type="checkbox"/>	i-015628497df6338a4	Web Server Qaziwa 2	80	us-east-1b	healthy	
<input type="checkbox"/>	i-0eec8b9304eb2e77e	Web Server Qaziwa 1	80	us-east-1a	healthy	

Health check settings

Protocol	Path	Port	Healthy threshold
HTTP	/	Traffic port	4 consecutive health check successes
Unhealthy threshold	Timeout	Interval	Success codes
3 consecutive health check failures	10 seconds	30 seconds	302

After you have created the target group, it should look like this:

Target groups (1/1) [Info](#)

Search or filter target groups

Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
<input checked="" type="checkbox"/> QaziwaTargetGroup	arn:aws:elasticloadbalancing:us-east-1:150114445082:targetgroup/QaziwaTargetGroup/b99f92d233fa05a	80	HTTP	Instance	QaziwaLoadBalancer	vpc-090a8ad552bd7ed6

[Create target group](#)

[Details](#) Targets Monitoring Health checks Attributes Tags

Details

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC vpc-090a8ad552bd7ed6
Load balancer QaziwaLoadBalancer			
Total targets 2	Healthy 2	Unhealthy 0	Unused 0
	Initial 0		Draining 0

And the load balancer should look like this:

Create Load Balancer Actions

Filter by tags and attributes or search by keyword

Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monitoring
<input checked="" type="checkbox"/> QaziwaLoadBalancer	QaziwaLoadBalancer-18874...	Active	vpc-090a8ad552bd7ed6	us-east-1a, us-east-1b	application	June 23, 2021 at 4:59:55 PM...	Edit

Load balancer: [QaziwaLoadBalancer](#)

Description Listeners Monitoring Integrated services Tags

Basic Configuration

Name	QaziwaLoadBalancer
ARN	arn:aws:elasticloadbalancing:us-east-1:150114445082:loadbalancer/app/QaziwaLoadBalancer/3564300b2f579c92
DNS name	QaziwaLoadBalancer-1887492094.us-east-1.elb.amazonaws.com (A Record)
State	Active
Type	application
Scheme	internet-facing
IP address type	IPv4
VPC	vpc-090a8ad552bd7ed6
Availability Zones	subnet-0fcac1e19b2de0e5 - us-east-1a (IPV4 address: Assigned by AWS) subnet-0df63aaac151cafb - us-east-1b (IPV4 address: Assigned by AWS)
Hosted zone	Z3SXDOOTRQ7XK
Creation time	June 23, 2021 at 4:59:55 PM UTC+0

Security

Security groups	sg-098d7a7a49643b7d, Qaziwa public security grp • Qaziwa public security grp
	Edit security groups

After configuring, we need to enable stickiness so that the session of the user on the instance will be consistent.

Go to target group > select the target group name > click on actions > edit attributes > check on the stickiness section

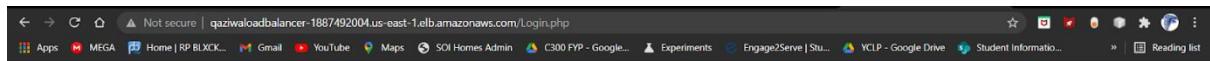
Edit attributes

Attributes	Restore defaults
<p>Deregistration delay The time to wait for in-flight requests to complete while deregistering a target. During this time, the state of the target is draining.</p> <p><input type="text" value="300"/> seconds 0-3600</p>	
<p>Slow start duration During this period, a newly registered target receives an increasing share of requests, until it reaches its fair share.</p> <p><input type="text" value="0"/> seconds Requires 30 to 900 seconds to enable, or 0 seconds to disable. This attribute cannot be combined with the Least outstanding requests algorithm.</p>	
<p>Load balancing algorithm Determines how the load balancer selects targets from this target group when routing requests.</p> <p><input checked="" type="radio"/> Round robin <input type="radio"/> Least outstanding requests Cannot be combined with the Slow start duration attribute.</p> <p><input checked="" type="checkbox"/> Stickiness The type of stickiness associated with this target group. If enabled, the load balancer binds a client's session to a specific instance within the target group.</p> <p>Stickiness type <input checked="" type="radio"/> Load balancer generated cookie <input type="radio"/> Application-based cookie</p> <p>Stickiness duration</p> <p><input type="text" value="3"/> <input type="text" value="days"/> ▾ 1 second - 7 days</p>	

[Cancel](#) [Save changes](#)

Targets	Monitoring	Health checks	Attributes	Tags												
<p>Attributes</p> <p>Edit</p>																
<table border="1"><tbody><tr><td>Stickiness</td><td>Deregistration delay</td></tr><tr><td>Enabled</td><td>300 seconds</td></tr><tr><td>Stickiness type</td><td>Stickiness duration</td></tr><tr><td>lb_cookie</td><td>1 day</td></tr><tr><td>Slow start duration</td><td>Load balancing algorithm</td></tr><tr><td>0 seconds</td><td>Round robin</td></tr></tbody></table>				Stickiness	Deregistration delay	Enabled	300 seconds	Stickiness type	Stickiness duration	lb_cookie	1 day	Slow start duration	Load balancing algorithm	0 seconds	Round robin	
Stickiness	Deregistration delay															
Enabled	300 seconds															
Stickiness type	Stickiness duration															
lb_cookie	1 day															
Slow start duration	Load balancing algorithm															
0 seconds	Round robin															

Testing of the Load Balancer DNS to see if it appear the Qaziwa Website:



Once we have done and tested that we are able to access the website, now is time to implement auto-scaling which helps to provide scalability and availability if there 1 or more instances are down. It can also control the number of instances created based on the traffic intensity, so it will suit best based on the traffic flow during the period of time.

Creation of Auto Scaling:

This function helps to ensure that there are steady resources available to every group at all times regardless of user demands. It works by monitoring the application and automatically scales the groups of different resources by adding or removing required capacity as the demand changes in real-time. This is part of ensuring maximum effectiveness within the application where high-demand groups have the necessary capacity to maintain the quality of service where low-demand groups will have capacity reduced for prevention of overspending.

There are various benefits of using auto-scaling in our application. A benefit is having better fault tolerance which will detect an unhealthy instance. This instance is then terminated and replaced with a new healthy instance. Knowing that auto-scaling ensures steadiness of resources, it has better availability to ensure all applications have the suitable capacity in handling the current traffic demand. In our application, auto-scaling is implemented to bridge 2 web server instances in 2 different public subnets from different availability zones. During our configuration, we will need to enable monitoring for auto-scaling to work. The main server is “Web Server Qaziwa 1” where all users will be directed to this server. Once the memory is being fully utilized (1 GiB) from the user traffic, any new users connecting to the website will be directed to the secondary web server “Web Server Qaziwa 2”. This is a technique where users will not experience server crashes due to surge of demand.

Firstly, we need to create an AMI (Amazon Machine Images) before we can launch auto-scaling. To do that we need to go to EC2 > instances > select an instance (Web Server Qaziwa 1) > Click on actions > go to Image and templates > Create Image

Create image [Info](#)

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

Instance ID
 [i-0eec8b9304eb2e77e](#) (Web Server Qaziwa 1)

Image name

Maximum 127 characters. Can't be modified after creation.

Image description - optional

Maximum 255 characters

No reboot
 Enable

Instance volumes

Volume type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/s...	Create new snapshot fr...	8	EBS General Purpose SS...	100		<input type="checkbox"/> Enable	<input checked="" type="checkbox"/> Enable

[Add volume](#)

ⓘ During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Tag image and snapshots together
Tag the image and the snapshots with the same tag.

Tag image and snapshots separately
Tag the image and the snapshots with different tags.

No tags associated with the resource.

After already fill up the requirements, then we can proceed to create an image

CloudWatch Metrics

Metrics Overview

Metrics

QaziwaWebServer

Value: 1

Time Range: June 24, 2021 to now

After it is active, we can proceed to create a launch configuration

In launch configuration,

Goto creates launch configuration > Name (QaziwaWebServerLaunchConfig) → AMI (QaziwaWebServer) → Instance type (t2.micro) → Monitoring (Enable EC2 monitoring) → Security group (Choose Qaziwa Public Security Group) → Choose our existing key (QaziwaKey).

Create launch configuration Info

Launch configuration name

Name

QaziwaWebServerLaunchConfig

Amazon machine image (AMI) Info

AMI

QaziwaWebServer



Instance type Info

Instance type

t2.micro (1 vCPUs, 1 GiB, EBS Only)

[Choose instance type](#)

Additional configuration - *optional*

Purchasing option Info

 Request Spot Instances

IAM instance profile Info

Select IAM role



Monitoring Info

 Enable EC2 instance detailed monitoring within CloudWatch

EBS-optimized instance

 Launch as EBS-optimized instance

► Advanced details

Later, if you want to use a different launch configuration, you can create a new one and apply it to any Auto Scaling group. Existing launch configurations cannot be edited.

Storage (volumes) [Info](#)

EBS volumes

Volume type	Devices	Snapshot	Size (GiB)	Volume type
Root	/dev/sda1	snap-0ca6666bdda6f3c1f	8	General purpose SSD

[+ Add new volume](#)

ⓘ Free tier eligible customers can get up to 30 GB of EBS storage. [Learn more about free usage tier eligibility and usage restrictions.](#)

Security groups [Info](#)

Assign a security group

Create a new security group
 Select an existing security group

Security groups

Security group ID	Name	VPC ID	Description
sg-0373078b58da83a2d	launch-wizard-1	vpc-090a8ad552bdc7ed6	launch-wizard-1 created 2021-06-18T12:16:53.142+08:00
sg-03c65391d4e841f9a	Database Test Server	vpc-090a8ad552bdc7ed6	launch-wizard-1 created 2021-06-18T11:31:02.982+08:00
sg-0aba88ff4c2c181fc	Qaziwa private security grp	vpc-090a8ad552bdc7ed6	Qaziwa private security grp
sg-0b1246adf0b167c3c	default	vpc-090a8ad552bdc7ed6	default VPC security group
sg-0ba7db7eb33d30e50	Qaziwa NAT security grp	vpc-090a8ad552bdc7ed6	launch-wizard-1 created 2021-06-02T15:53:41.005+08:00
sg-0d1089b691aab8a9f	default	vpc-022432903ae41160e	default VPC security group
sg-0f6bd7a7a49643b7d	Qaziwa public security grp	vpc-090a8ad552bdc7ed6	Qaziwa public security grp

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Key pair (login) [Info](#)

Key pair options

Choose an existing key pair

Existing key pair

QaziwaKey

I acknowledge that I have access to the selected private key file (QaziwaKey.pem), and that without this file, I won't be able to log into my instance.

[Cancel](#) [Create launch configuration](#)

After the configuration, then click on “create launch configuration”.

Now is the time to configure autoscaling

Goto Autoscaling > Create Auto Scaling Group > Auto Scaling Name (QaziwaAutoScaling) → Launch Configuration (QaziwaWebServerLaunchConfig) [taken from Launch Configuration Group > click next

Step 1
Choose launch template or configuration

Step 2
Configure settings

Step 3 (optional)
Configure advanced options

Step 4 (optional)
Configure group size and scaling policies

Step 5 (optional)
Add notifications

Step 6 (optional)
Add tags

Step 7
Review

Choose launch template or configuration

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

Name

Auto Scaling group name
Enter a name to identify the group.
QaziwaAutoScaling

Must be unique to this account in the current Region and no more than 255 characters.

Launch configuration [Info](#) [Switch to launch template](#)

Launch configuration
Choose a launch configuration that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

QaziwaWebServerLaunchConfig

Create a launch configuration ?	Launch configuration QaziwaWebServerLaunchConfig	AMI ID ami-0db2e12111c346da2	Date created Thu Jun 24 2021 17:10:05 GMT+0800 (Singapore Standard Time)
Security groups sg-0f6bd7a7a49643b7d ?	Instance type t2.micro	Key pair name -	

Cancel **Next**

In the configure Setting → Select a VPC (Qaziwa VPC) → Subnets (Public Subnet 1 & 2) > click next

Step 1
Choose launch template or configuration

Step 2
Configure settings

Step 3 (optional)
Configure advanced options

Step 4 (optional)
Configure group size and scaling policies

Step 5 (optional)
Add notifications

Step 6 (optional)
Add tags

Step 7
Review

Configure settings [Info](#)

Configure the settings below. Depending on whether you chose a launch template, these settings may include options to help you make optimal use of EC2 resources.

Network [Info](#)

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC

vpc-090a8ad552bdc7ed6 (Qaziwa VPC) 10.0.0.0/16 [?](#)

Create a VPC [?](#)

Subnets

Select subnets [?](#)

us-east-1a | subnet-0bcca6218d2bed2e5 (Qaziwa public subnet) 10.0.1.0/24 [X](#)

us-east-1b | subnet-0d563aaca151ca8f8 (Qaziwa public subnet 2) 10.0.3.0/24 [X](#)

Create a subnet [?](#)

Cancel **Previous** **Skip to review** **Next**

In the configure advanced options, Click on Attach to an existing load balancer → choose from your load balancer target group → choose the existing load balancer target group (QaziwaTargetGroup) → in the additional settings → Enable group metrics collection within cloudwatch > click next

Step 1
[Choose launch template or configuration](#)

Step 2
[Configure settings](#)

Step 3 (optional)
[Configure advanced options](#)

Step 4 (optional)
[Configure group size and scaling policies](#)

Step 5 (optional)
[Add notifications](#)

Step 6 (optional)
[Add tags](#)

Step 7
[Review](#)

Configure advanced options Info

Choose a load balancer to distribute incoming traffic for your application across instances to make it more reliable and easily scalable. You can also set options that give you more control over health check replacements and monitoring.

Load balancing - optional Info

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

- No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.
- Attach to an existing load balancer
Choose from your existing load balancers.
- Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.

Attach to an existing load balancer

Select the load balancers that you want to attach to your Auto Scaling group.

- Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.
- Choose from Classic Load Balancers

Existing load balancer target groups
Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups

QaziwaTargetGroup | HTTP

X

QaziwaTargetGroup | HTTP
X

Health checks - optional

Health check type Info
EC2 Auto Scaling automatically replaces instances that fail health checks. If you enabled load balancing, you can enable ELB health checks in addition to the EC2 health checks that are always enabled.

EC2 ELB

Health check grace period
The amount of time until EC2 Auto Scaling performs the first health check on new instances after they are put into service.

300

seconds

Additional settings - optional

Monitoring Info

Enable group metrics collection within CloudWatch

Cancel
Previous
Skip to review
Next

In the configure group size and scaling policies, fill it accordingly > click next

Step 1
Choose launch template or configuration

Step 2
Configure settings

Step 3 (optional)
Configure advanced options

Step 4 (optional)
Configure group size and scaling policies

Step 5 (optional)
Add notifications

Step 6 (optional)
Add tags

Step 7
Review

Configure group size and scaling policies Info

Set the desired, minimum, and maximum capacity of your Auto Scaling group. You can optionally add a scaling policy to dynamically scale the number of instances in the group.

Group size - optional Info

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity

Minimum capacity

Maximum capacity

Scaling policies - optional

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. Info

Target tracking scaling policy
Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

None

Instance scale-in protection - optional

Instance scale-in protection
If protect from scale in is enabled, newly launched instances will be protected from scale in by default.

Enable instance scale-in protection

Cancel Previous Skip to review Next

Click next for notifications and add tags and on the review page, click on create auto scaling group

EC2 > Auto Scaling groups

Auto Scaling groups (1/1)

Search your Auto Scaling groups

Name	Launch template/configuration	Instances	Status	Desired capacity	Min.	Max.	Availability Zones
QaziwaAutoScaling	QaziwaWebServerLaunchConfig	0	Updating capacity	2	2	3	us-east-1a, us-east-1b

Details Activity Automatic scaling Instance management Monitoring Instance refresh

Group details

Desired capacity 2	Auto Scaling group name QaziwaAutoScaling
Minimum capacity 2	Date created Thu Jun 24 2021 17:30:52 GMT+0800 (Singapore Standard Time)
Maximum capacity 3	Amazon Resource Name (ARN) arn:aws:autoscaling:us-east-1:50114445082:autoScalingGroup:6f486db3-0d81-4ae9-a438-b5d8f00b8590:autoScalingGroupName/QaziwaAutoScaling

Launch configuration

Launch configuration QaziwaWebServerLaunchConfig	AMI ID ami-0db2e12111c346da2	Security groups sg-0f6bd7a7a09643b7d
---	---------------------------------	---

The screenshot shows the AWS Auto Scaling Groups page. At the top, there is a search bar and a table header with columns: Name, Launch template/configuration, Instances, Status, Desired capacity, and Availability Zones. A row for 'QaziwaAutoScaling' is selected, showing its status as 'Updating capacity' with a value of 2. Below the table, there are sections for 'Notifications' and 'Activity history'. The 'Activity history' section lists two events: one successful instance launch and one pre-service instance launch. The bottom part of the screenshot shows the 'Instance management' tab selected, displaying a table of instances with columns: Instance ID, Lifecycle, Instance type, Weighted capacity, Launch template/configuration, Availability Zone, Health status, and Protected from. Two instances are listed: one in us-east-1a and one in us-east-1b, both marked as healthy.

The above shows that we have successfully created an Auto Scaling group.

Since all of the configurations are stable and able to access the website and provide scalability, functional, stress tests that meet our requirements, we then need to set up a domain name for our website.

Creation of Domain Name:

Freenom Domain:

In normal real-life usage, websites have domain names for easy identification among normal users.

We will be using freenom which is a free DNS Website that enables us to get our own DNS free for 3 months. (<https://www.freenom.com/en/index.html?lang=en>)

To start off, we check if our preferred DNS (www.qaziwa.tk) is available.

The screenshot shows the freenom website's domain search results. A search bar at the top contains 'qaziwa.tk'. To its right is a blue circular button labeled 'Check Availability'. Below the search bar, a message says 'Yes qaziwa.tk is available!' in blue. To the left, the domain name 'qaziwa' is shown with a blue badge that says '.FREE'. To the right, the price is listed as 'USD 0.00' and there is a green button with a checkmark that says 'Selected'. At the bottom right of the main content area, there is a green 'Checkout' button.

Yes, it is available and it is free, so now we are going to proceed to have this domain up and ready.

Click checkout

A screenshot of a search results page. At the top right, it says 'Showing results 1 / 20' and has a 'SHOW ALL' button. Below that is a 'SHOW MORE' button. To the right, there is a green 'Checkout' button. Both the '1 domain in cart' text and the 'Checkout' button are circled in red.

Click on Continue

A screenshot of a domain configuration page. It shows a table with columns for 'Domain', 'IDSHIELD' (with a shield icon), 'Use your new domain' (with a gear icon), and 'Period'. The domain 'qaziwa.tk' is listed. Below the table are buttons for 'Forward this domain' and 'Use DNS'. To the right, the period is set to '3 Months @ FREE'. At the bottom right is a large blue 'Continue' button.

Moving forward, we sign in using our FYP Gmail account and password

 Sign in with Google

Sign in

to continue to [freenom.com](#)

Email or phone —
fypc300qaziwa@gmail.com

[Forgot email?](#)

[Create account](#) [Next](#)

Then verify our email and then fill in our credentials

Your Details

First Name	FYP
Last Name	C300
Company Name	AWS Cloud Implementation and Security
Address 1	NIL
Zip Code	NIL
City	NIL
Country	Singapore
State/Region	North East ▾
Phone Number	+65 [REDACTED]
Email Address	fypc300qaziwa@gmail.com Change

Tax may be charged depending upon the state and country selections you make. Click to recalculate after making your choices. [Update](#)

Then click checkout.

Now we have our domain up and ready

Enter Domain to Find					Filter
Domain	Registration Date	Expiry date	Status	Type	
qaziwa.tk	2021-06-24	2021-09-24	ACTIVE	Free	Manage Domain
Results Per Page: 10 ▾					
1 Records Found, Page 1 of 1					

Then we need to configure our DNS record so that we are able to access the website using the load balancer DNS

Go to Manage domain > Manage freenom DNS > do the configuration shown below:

[« Back to domain details](#)

Record added successfully

Modify Records

Name	Type	TTL	Target	Action
WWW	CNAME	3600	qaziwaloadbalancer-1887492004.us-east-1.elb.amazonaws.com	Delete

[Add Records](#) [Save Changes](#)

Now is to test and check

Domain Dossier Investigate domains and IP addresses

domain or IP address

domain whois record DNS records traceroute

network whois record service scan [go](#)

user: anonymous [202.21.159.198]
balance: 40 units
[log in](#) | [account info](#)

[CentralOps.net](#)

Do you see Whois records that are missing contact information?
[Read about reduced Whois data due to the GDPR.](#)

Address lookup

canonical name [qaziwaloadbalancer-1887492004.us-east-1.elb.amazonaws.com](#).
aliases [www.qaziwa.tk](#)
addresses [54.235.157.146](#)
[52.200.236.23](#)

DNS records

name	class	type	data	time to live
www.qaziwa.tk	IN	CNAME	qaziwaloadbalancer-1887492004.us-east-1.elb.amazonaws.com	3600s (01:00:00)
qaziwaloadbalancer-1887492004.us-east-1.elb.amazonaws.com	IN	A	52.200.236.23	60s (00:01:00)
qaziwaloadbalancer-1887492004.us-east-1.elb.amazonaws.com	IN	A	54.235.157.146	60s (00:01:00)
us-east-1.elb.amazonaws.com	IN	NS	ns-1119.awsdns-11.org	1800s (00:30:00)
us-east-1.elb.amazonaws.com	IN	NS	ns-1793.awsdns-32.co.uk	1800s (00:30:00)
us-east-1.elb.amazonaws.com	IN	NS	ns-235.awsdns-29.com	1800s (00:30:00)
us-east-1.elb.amazonaws.com	IN	NS	ns-934.awsdns-52.net	1800s (00:30:00)
us-east-1.elb.amazonaws.com	IN	SOA	server: ns-1119.awsdns-11.org email: awsdns-hostmaster@amazon.com serial: 1 refresh: 7200 retry: 900 expire: 1209600 minimum ttl: 60	60s (00:01:00)
qaziwa.tk	IN	SOA	server: ns01.freenom.com email: soa@freenom.com serial: 1624523411 refresh: 10800 retry: 3600 expire: 604800 minimum ttl: 3600	300s (00:05:00)
qaziwa.tk	IN	NS	ns02.freenom.com	300s (00:05:00)
qaziwa.tk	IN	NS	ns03.freenom.com	300s (00:05:00)
qaziwa.tk	IN	NS	ns01.freenom.com	300s (00:05:00)
qaziwa.tk	IN	NS	ns04.freenom.com	300s (00:05:00)
146.157.235.54.in-addr.arpa	IN	PTR	ec2-54-235-157-146.compute-1.amazonaws.com	300s (00:05:00)
157.235.54.in-addr.arpa	IN	SOA	server: dns-external-master.amazon.com email: root@amazon.com serial: 14 refresh: 3600 retry: 900 expire: 604800 minimum ttl: 900	900s (00:15:00)



QaZiWa Books - Login



Login

Email Address:

Password:

[Sign-up for free!!](#)

QaZiWa Copyright
© 2021

So now, after waiting for a few minutes, we can see that our Domain is up and running. As shown above, we can see that the connection is not secure, therefore more research is needed to ensure that we have a secure SSL Certificate and connection from the client to the load balancer.

Creation of CloudFlare:

Cloudflare SSL Certificate Configuration

It is important to ensure that having an SSL Certificate to be implemented on the website. SSL Certificate is a type of cryptography which is also known as a digital certificate that enables electronic documents such as websites to be encrypted when transmitting its data. This means all of the private information such as user information will be encrypted and this causes the information not to be in plain text. As such, implementing an SSL Certificate will

provide good data in transmit encryption to the AWS load balancer. For the SSL Certificate, we will be using a CloudFlare website, which is a free security website that provides multiple features such as DNS, Firewall, Caching, and most importantly SSL/TLS encryption. The link to the website (<https://www.cloudflare.com>)

Firstly u need to sign in to the web service with your credentials

Introducing Zero Trust Private Networking

Replace your VPN with identity-based Zero Trust policies inside your private network. Now you can lock down access to corporate applications, private IP spaces and hostnames. Free for up to 50 users.

[Learn More](#)



After login, add your DNS into the add a site section → qaziwa.tk

Please verify your email address to receive updates and notices for your account.

qaziwa.tk

+ Add a Site

Search websites in Wafimuhd5@gmail.com's Account...

qaziwa.tk
✓ Active

Create custom billing alerts.
Receive an email when billable usage exceeds a certain amount. ⓘ
Configure

After adding our DNS to the Cloudflare website, then it will show Cloudflare nameservers

Cloudflare nameservers

To use Cloudflare, ensure your authoritative DNS servers, or nameservers have been changed. These are your assigned Cloudflare nameservers.

Type	Value
NS	pablo.ns.cloudflare.com
NS	serena.ns.cloudflare.com

Replace the name server from the freenom to the nameserver of the Cloudflare

Managing qaziwa.tk

Information Upgrade Management Tools ▾ Manage Freenom DNS

Nameservers

You can change where your domain points to here.
Please be aware changes can take up to 24 hours to propagate.

Use default nameservers (Freenom Nameservers)

Use custom nameservers (enter below)

Nameserver 1

PABLO.NS.CLOUDFLARE.COM

Nameserver 2

SERENA.NS.CLOUDFLARE.COM

Nameserver 3

After this go back to the cloudflare website and then you will see this

Type	Name	Content	TTL	Proxy status
CNAME	www	qaziwaloadbalancer-1887492004....	Auto	Proxied

I have enabled proxied, in which it will hide the public IP address of the load balancer and instead it will display the cloudflare IP address which hides the true IP address of the Load Balancer.

```
C:\Users\19031563>ping www.qaziwa.tk

Pinging www.qaziwa.tk [104.21.87.193] with 32 bytes of data:
Reply from 104.21.87.193: bytes=32 time=6ms TTL=58
Reply from 104.21.87.193: bytes=32 time=6ms TTL=58
Reply from 104.21.87.193: bytes=32 time=8ms TTL=58
Reply from 104.21.87.193: bytes=32 time=6ms TTL=58

Ping statistics for 104.21.87.193:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 6ms, Maximum = 8ms, Average = 6ms
```

```
C:\Users\19031563>nslookup www.qaziwa.tk
Server:  router.asus.com
Address:  192.168.50.1

Non-authoritative answer:
Name:    www.qaziwa.tk
Addresses:  2606:4700:3035::6815:57c1
          2606:4700:3035::ac43:91b4
          104.21.87.193
          172.67.145.180

C:\Users\19031563>
```

Domain Dossier Investigate domains and IP addresses

domain or IP address

domain whois record DNS records traceroute

network whois record service scan

user: anonymous [180.129.48.35]
balance: 48 units
[log in](#) | [account info](#)

CentralOps.net

Do you see Whois records that are missing contact information?
[Read about reduced Whois data due to the GDPR.](#)

Address lookup

canonical name [www.qaziwa.tk](#).

aliases

addresses **104.21.87.193**
172.67.145.180
2606:4700:3035::ac43:91b4
2606:4700:3035::6815:57c1

DNS records

name	class	type	data	time to live
www.qaziwa.tk	IN	HINFO	CPU: RFC8482 OS:	3789s (01:03:09)
qaziwa.tk	IN	HINFO	CPU: RFC8482 OS:	3789s (01:03:09)
qaziwa.tk	IN	NS	pablo.ns.cloudflare.com	86400s (1.00:00:00)
qaziwa.tk	IN	NS	serena.ns.cloudflare.com	86400s (1.00:00:00)
193.87.21.104.in-addr.arpa	IN	HINFO	CPU: RFC8482 OS:	3789s (01:03:09)
21.104.in-addr.arpa	IN	NS	cruz.ns.cloudflare.com	53281s (14:48:01)
21.104.in-addr.arpa	IN	NS	kevin.ns.cloudflare.com	53281s (14:48:01)
4.b.19.3.4.c.a.0.0.0.0.0.0.0.0.0.0.5.3.0.3.0.7.4.6.0.6.2.ip6.arpa	IN	HINFO	CPU: RFC8482 OS:	3789s (01:03:09)
0.0.7.4.6.0.6.2.ip6.arpa	IN	NS	leah.ns.cloudflare.com	54683s (15:11:23)
0.0.7.4.6.0.6.2.ip6.arpa	IN	NS	micah.ns.cloudflare.com	54683s (15:11:23)

Service scan

FTP - 21 Error: TimedOut
SMTP - 25 Error: TimedOut

And we tally with the AWS Load Balancer Public IP address it shows

Network interfaces (2) Info						
script	Instance ID	Status	Public IPv4 address	Primary private IPv4 address	Secondary private IPv4 ad...	Actions
3 app/QaziwaLoadB...	-	In-use	34.238.111.255	10.0.3.41	-	Edit Delete
3 app/QaziwaLoadB...	-	In-use	52.200.236.23	10.0.1.22	-	Edit Delete

Therefore, we can conclude that it hides the IP address of the actual load balancer IP address and therefore, users cannot find the actual load balancer IP address.

Moving forward, now we want to enable SSL Certificate, go to SSL/TLS > check on flexible → which enable it to encrypt all of the data from user or browser to cloudflare.

The screenshot shows the Cloudflare dashboard with the SSL/TLS section selected. At the top, there's a navigation bar with icons for Overview, Analytics, DNS, SSL/TLS (which is highlighted), Firewall, Access, Speed, Caching, Workers, Rules, Network, Traffic, Stream, Custom Pages, Apps, and Scrape Shield. Below the navigation bar, there are tabs for Overview, Edge Certificates, Client Certificates, Origin Server, and Custom Hostnames. The Overview tab is currently active. In the main content area, there's a message: "Your SSL/TLS encryption mode is Flexible". It says this setting was last changed 34 minutes ago. To the right, there are four radio button options: "Off (not secure)" (disabled), "Flexible" (selected, indicated by a blue outline), "Full" (disabled), and "Full (strict)" (disabled). A diagram illustrates the setup: a "Browser" icon is connected to a "Cloudflare" icon (which is highlighted with a red circle), which is then connected to an "Origin Server" icon. Below the diagram, there's a link to "Learn more about End-to-end encryption with Cloudflare".

Due to load balancer limitation and less privilege/not authorised to do an SSL Certificate, therefore we cannot import a certificate that is provided by cloudflare to the load balancer. After this setting now we want to tick on the Always Use HTTPS section shown below.

Overview Edge Certificates Client Certificates Origin Server Custom Hostnames

Edge Certificates

Manage and purchase SSL certificates that will be served to your web visitors.

Your plan includes a shared Cloudflare Universal SSL certificate. To get a dedicated certificate with custom hostnames [place a certificate order](#).

Your plan does not allow you to upload any SSL certificates, but you may [order an auto-renewing certificate](#) or [upgrade](#) to the Business plan to enable this feature.

Hosts	Type	Status	Expires on
*.qaziwa.tk, qaziwa.tk	Universal	Active	2021-09-24 (Managed) ▶

◀ ▶ 1–1 of 1 certificates

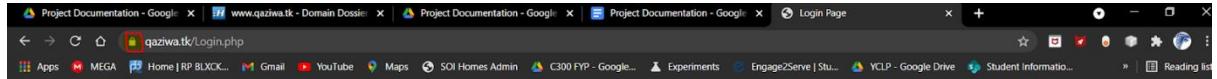
[API](#) ▶ [Help](#) ▶

Always Use HTTPS

Redirect all requests with scheme "http" to "https". This applies to all http requests to the zone.

On ▶

After all of this, we wait for a moment, and then we can see that our new website is already used HTTPS website and not HTTP



QaZiWa Books - Login



Login

Email Address:

Password:

[Sign-up for free!!](#)

QaZiWa Copyright
© 2021



QaZiWa Books - Login



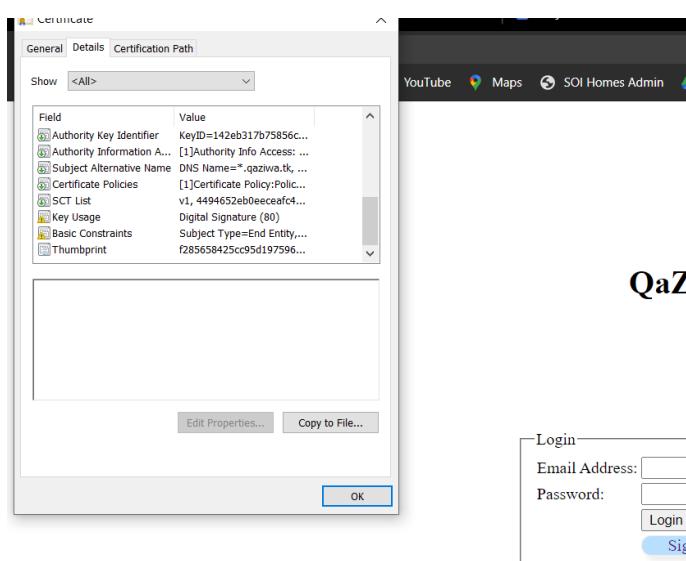
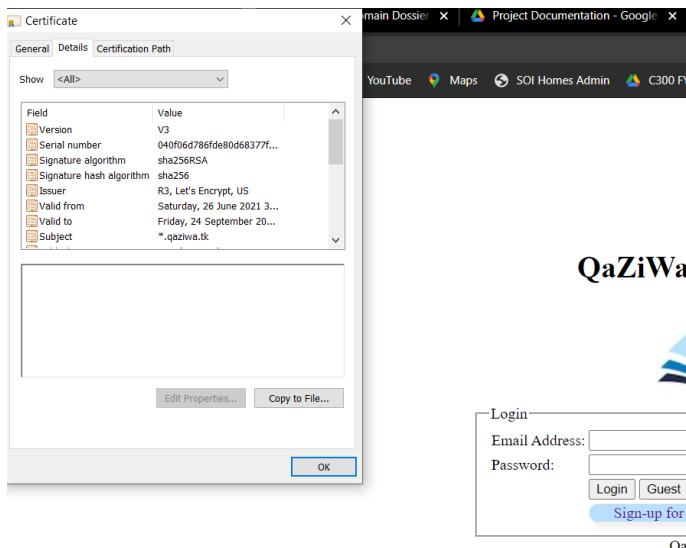
Login

Email Address:

Password:

[Sign-up for free!!](#)

QaZiWa Copyright
© 2021



Now we have fully secured the website connection from the user to the cloudflare web service.

Allowing Load Balancer to accept traffic from Cloudflare:

However, there is an issue doing this. If the user finds out the ALB DNS name, he or she can access the load balancer directly in which it does not pass through cloudflare and this can cause a security risk as an attacker can do a DoS attack on our load balancer. To prevent this, we created a security group specifically on access to the Cloudflare IP address.

Below is a security group for the CloudFlare IP addresses.



list of cloudflare ips

173.245.48.0/20
 103.21.244.0/22
 103.22.200.0/22
 103.31.4.0/22
 141.101.64.0/18
 108.162.192.0/18
 190.93.240.0/20
 188.114.96.0/20
 197.234.240.0/22
 198.41.128.0/17
 162.158.0.0/15
 172.64.0.0/13
 131.0.72.0/22
 104.16.0.0/13
 104.24.0.0/14

The above is the IP address of the cloudflare, so this means any of this IP must be added to the Load Balancer Security Group. So since we already know the security group of the cloudflare, now we need to create a security group to allow only these IP addresses from the cloudflare.

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	173.245.48.0/20	-
HTTP	TCP	80	103.21.244.0/22	-
HTTP	TCP	80	103.22.200.0/22	-
HTTP	TCP	80	103.31.4.0/22	-
HTTP	TCP	80	141.101.64.0/18	-
HTTP	TCP	80	108.162.192.0/18	-
HTTP	TCP	80	190.93.240.0/20	-
HTTP	TCP	80	188.114.96.0/20	-
HTTP	TCP	80	197.234.240.0/22	-
HTTP	TCP	80	198.41.128.0/17	-
HTTP	TCP	80	162.158.0.0/15	-
HTTP	TCP	80	172.64.0.0/13	-
HTTP	TCP	80	131.0.72.0/22	-
HTTP	TCP	80	104.16.0.0/13	-
HTTP	TCP	80	104.24.0.0/14	-

Inbound rules (15)				
Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	173.245.48.0/20	–
HTTP	TCP	80	103.21.244.0/22	–
HTTP	TCP	80	103.22.200.0/22	–
HTTP	TCP	80	103.31.4.0/22	–
HTTP	TCP	80	141.101.64.0/18	–
HTTP	TCP	80	108.162.192.0/18	–
HTTP	TCP	80	190.93.240.0/20	–
HTTP	TCP	80	188.114.96.0/20	–
HTTP	TCP	80	197.234.240.0/22	–
HTTP	TCP	80	198.41.128.0/17	–
HTTP	TCP	80	162.158.0.0/15	–
HTTP	TCP	80	172.64.0.0/13	–
HTTP	TCP	80	131.0.72.0/22	–
HTTP	TCP	80	104.16.0.0/13	–
HTTP	TCP	80	104.24.0.0/14	–

Above, it shows that we have already created a security group that only allows IP addresses from the selected source IP addresses with port 80 (HTTP). This is by, we want to access a website, hence port 80 is being selected. After the creation of the security group and assigned the specific IP addresses. Now we need to integrate it into our LB. To do this we need to go to Load Balancer, select Qazia LoadBalancer then click on actions, and edit the security groups that we have created.

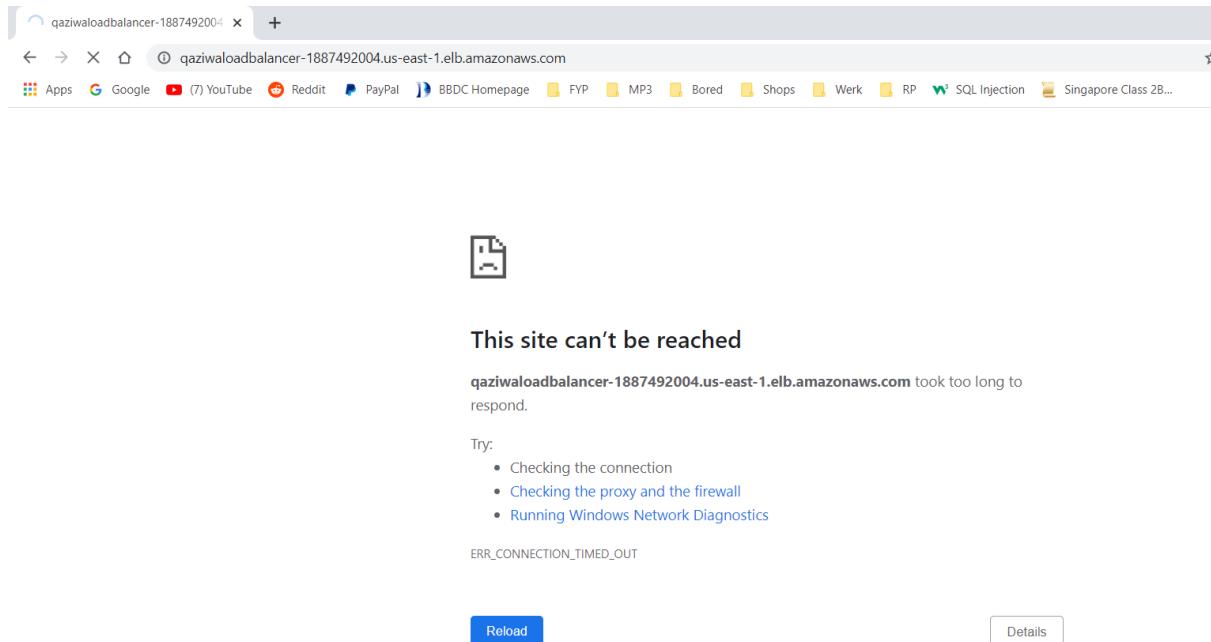
Edit security groups X

Select security groups to associate with your load balancer.

<input type="checkbox"/> Security group ID	Name	Description
<input type="checkbox"/>	sg-03c65391d4e84...	Database Test Server
<input checked="" type="checkbox"/>	sg-079f5a8a62be5b...	Qaziwa CloudFlare ...
<input type="checkbox"/>	sg-0ba7db7eb33d3...	Qaziwa NAT secur...
<input type="checkbox"/>	sg-0aba88ff4c2c181fc	Qaziwa private sec...
<input type="checkbox"/>	sg-0f6bd7a7a49643...	Qaziwa public secur...
<input type="checkbox"/>	sg-0b1246adf0b167...	default

Cancel Save

After we have selected, then we click Save, then wait for a few seconds for it to take effect. Now we need to test if we are still able to access the Load Balancer DNS.



As shown above, we can see that we are unable to access the load balancer DNS, so now we test it to access the domain name that we have assigned to it which is www.qaziwa.tk



As shown above, we can see that we are able to access the website directly from the client to the cloudflare, but not from the client to the load balancer. This is an added feature to ensure that users are not able to access the load balancer directly and they must go through the cloudflare in order to access the website.

Allowing Instance to accept traffic from Load Balancer:

Going further, since we have secured the load balancer, now we want to secure the access of the IP address from the load balancer to the instance itself. This can be seen below.



QaZiWa Books - Login

Login

Email Address:

Password:

QaZiWa Copyright
© 2021

When accessing the IP address of the instance, we can still view the Web Page, so we need to edit the security group and assign the security to the instances. But before we add the security group, we need to know the IP address of the load balancer, so we need to go to network interfaces, and from network interfaces, we search for the load balancer and then find the IP address section.

	Public IPv4 address	Primary private IPv4 address	Seconda
	52.73.17.38	10.0.1.52	-
	52.20.226.103	10.0.3.203	-

Since there are 2 IP addresses Public and Private, we need to take the private subnet and add it into the security group.

Security Groups (1/9) Info						
	Name	Security group ID	Security group name	VPC ID	Description	Owner
<input type="checkbox"/>	Database Test Server	sg-03c65391d4e841f9a	Database Test Server	vpc-090a8ad552bcd7ed6	launch-wizard-1 create...	5011444
<input checked="" type="checkbox"/>	Qaziwa LoadBalancer IPs	sg-044f3ce12d2657282	Qaziwa Loadbalancer IPs	vpc-090a8ad552bcd7ed6	Allow Loadbalancer to ...	5011444

sg-044f3ce12d2657282 - Qaziwa Loadbalancer IPs

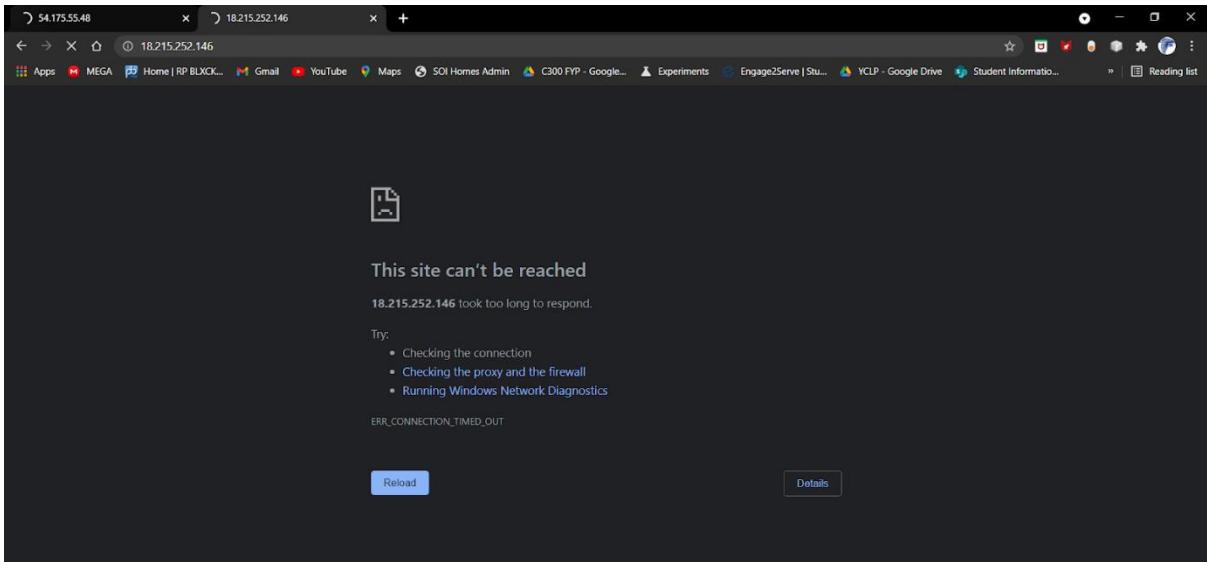
Details [Inbound rules](#) [Outbound rules](#) Tags

Inbound rules (2)

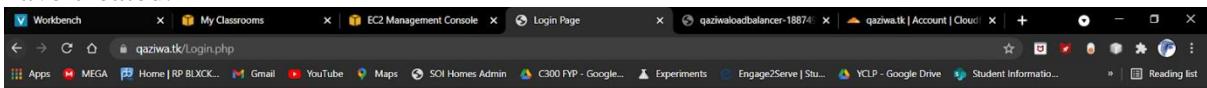
Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	10.0.1.52/32	-
HTTP	TCP	80	10.0.3.203/32	-

As shown above, we can see that the security group has been created and the rule is applied as HTTP due to it being web browsing and the private IP address of load balancer. 2 different IP addresses by 2 different zones. Now we need to stop the instance and then detach the old

security group to the new security group that we have created that only assigns to the IP address of the load balancer.



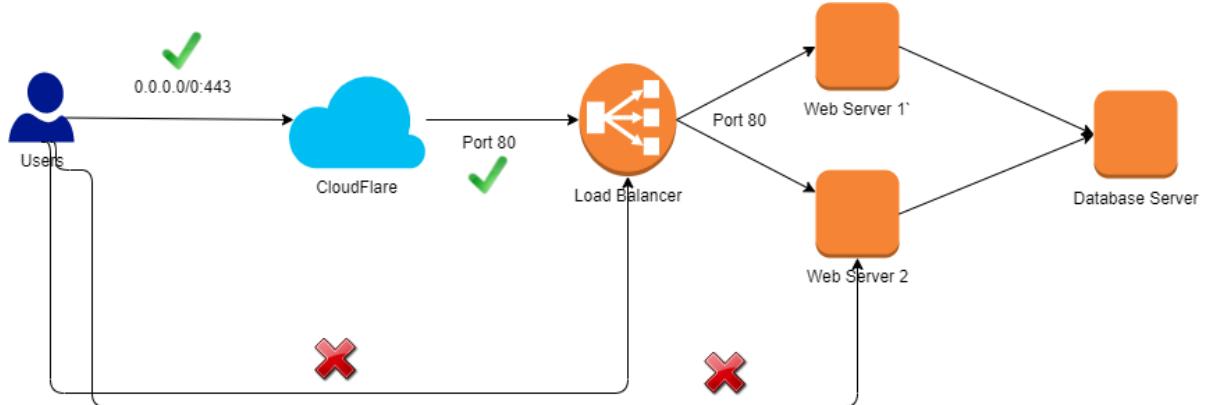
So when we access the instance directly, we can see that we are unable to access the Web Server itself, so therefore to only access this website is by using the domain name that we have created.



As shown above, you can see that when we access the domain itself, we are able to access the webpage and see its content. This is shown that the user cannot access any DNS of the load balancer and even the IP address of the Web Server. They can only access the domain name that is given. This is part of the security features so that the user needs to pass through the cloudflare which helps in detecting malicious content such as DoS, web attacks and much more and following to that, when they access the web server itself, the web server already got another level of Web application firewall and security in which it will also block any unwanted traffic or suspicious packets. Hence, we have applied defense in depth in our network.

Since we know that due to the AWS limitation of importing certificates and implementing it on LB, we cannot do it. As such after the cloudflare, all of the traffic will be unencrypted.

In summary, we have configured the traffic flow in such a way that users will not have the access to the load balancer or the instance directly, but instead they need to go through the cloudflare in order to access the website.



Implementation of SSL Certificate involving instance:

Since we have SSL encryption from the client to the cloudflare itself and not the load balancer (due to not enough privileges given in AWS educate account), we can see that there are vulnerabilities from the cloudflare to the load balancer as it is not encrypted. I pictures is shown below

✓ Your SSL/TLS encryption mode is Full (strict)

This setting was last changed an hour ago

Off (not secure) ⓘ
 No encryption applied

Flexible
 Encrypts traffic between the browser and Cloudflare

Full
 Encrypts end-to-end, using a self signed certificate on the server

Full (strict)
 Encrypts end-to-end, but requires a trusted CA or Cloudflare Origin CA certificate on the server

Learn more about [End-to-end encryption with Cloudflare](#)

The encrypted part is the Browser to the Cloudflare but not from cloudflare to the instance, therefore, what if an attacker was somehow between the cloudflare and the EC2 instance?

To answer, we will deep down on SSL encryption and ensure that we have a full (strict) SSL encryption.

Firstly, we need to ensure that the web server is built with defense in depth by implementing several configurations and installing mods such as mod security (refer to security testing). Firstly we settle that defense in depth hardening and also even install apache server in AWS EC2 instance. After settling with the web server configuration, now we need to assign an elastic IP to ensure that this ec2 instance has a static public IP address and it would not change (used from the Elastic IP section).

After assigning to the EC2 instance, now we need to create a domain for this particular instance by going to freenom and set up the configuration (refer to Domain name section), then after that, we assign it to cloudflare to manage our DNS names. For the DNS name, we name it qaziwa1.tk. Here, we enable the SSL certificate in the cloudflare to Full (Strict) section.

Your SSL/TLS encryption mode is Full (strict)

This setting was last changed an hour ago

Off (not secure) ⓘ
 No encryption applied

Flexible
 Encrypts traffic between the browser and Cloudflare

Full
 Encrypts end-to-end, using a self signed certificate on the server

Full (strict)
 Encrypts end-to-end, but requires a trusted CA or Cloudflare Origin CA certificate on the server

Learn more about [End-to-end encryption with Cloudflare](#)

API ▶ Help ▶

After we already put the SSL/TLS encryption mode to Full (strict), now we need to go to “Origin Server > Create Certificate > Generate private key and CSR with Cloudflare (RSA 2048)> set the variable shown below> click on create” Then from there you will see a certificate key and the private key. Then we need to copy it into our notepad.

[← Back](#)

Origin Certificate Installation

Follow the steps below to install a certificate on your origin server.

The first step in generating a certificate for your origin is creating a private key and a Certificate Signing Request (CSR). You can provide your own CSR or we can generate a key and CSR using your web browser.

- Generate private key and CSR with Cloudflare

Private key type

RSA (2048)

- Use my private key and CSR

List the hostnames (including wildcards) on your origin that the certificate should protect. By default your origin certificate covers the apex of your domain (`example.com`) and a wildcard (`*.example.com`). If there are others you wish to add, e.g., those not covered by the wildcard such as `one.two.example.com`, you can add them below.

Hostnames

*.qaziwa1.tk qaziwa1.tk

Choose how long before your certificate expires. By default your certificate will be valid for fifteen (15) years. If you'd like to decrease how long your certificate will be valid make a selection below.

Certificate Validity

15 years

[Cancel](#)

[Create](#)

```
-----BEGIN CERTIFICATE-----  
MIIEoDCCA4igAwIBAgIUP1Kd++jIMFsQB7zOEe7pvD+S  
BQAwgYsxCzAJBgNVBAYTA1VTMRkwFwYDVQQKExBDbG91  
MgYDVQQLEytDb691ZEzsYXJ1IE9yaWdpbiBTU0wgQ2Vy  
aXRSMRYwFAVDVQOHewLTW4grRnJhbmNpc2NvMRMwEQYC  
MB4XDITIxMDcwMzE4NDUwMFoXDTM2MDYyOTE4NDUwMFow  
dWRGbGFyZSwgSW5jLjEdMBsGA1UECxMUQ2xvdWRGbGFy  
BgnVBAMTHUNsb3VkRmxhcmUgT3JpZ2luIENlcnPzmlj  
9w0BAQEFAOCAsQAMIIBCgKCAQEAnml4ryO06D+hj8g+  
Sare6rDl3UiK6Bprm+SczNLfji8yTjojjhZxPSguOE1C  
GducVrIOf3ZgsV6AS12LIZTizVkp1ZFewbzk8TAhB0wj  
FgdidabC0qwAvz/mUNGu4WTbMuo8zktP9maFXsn8tq  
XNr0msejQFPQsoA/mJCqODCYzYocTbWleneY8/N2Ruh  
U62ZcG3v8An0NNvbrs4+A0gwORnCjbDnh0c4W0cdsfwI  
AR4wDgYDVR0PAQH/BAQDAgNgMB0GA1UDJQOWMBQGCCsC  
ATANBgNVHRMBAf8EAjAAAMB0GA1UdDgQWBBSOHR/K3tm4  
BgnVHSMEGDAwgbQk6FNXXXw0QIep65TbuuEWpWppDBA  
MAYIKwYBBQUHMAggJGh0dHA6Ly9vY3NwLnNsb3VkZmxh  
YTAjBgNVHREEHDaggwqLnFheml3YTEudGuCCnFheml3  
LzAtocugKYYnaHR0cDovL2Nybc5jbG91ZGzsYXJ1LmNv  
MA0GCSqGSIB3DQEBCwUA4IBAQc7mGNJTEZeKvMxVRrz  
Slvk+S81+3He7w0cpSJXB0zIMLxFH7QFB6usZPYn9rs  
pkJNVAg5itqKG13m/YLTZF7TsGnX476ol8w8L2xFbsB8  
4flhePDbeQVXzclpvGoNkGnIIrnA1W1wxnLSSVZ991Y  
qpvtkqyTPzxU+g06yx0cGxRnyuN2Q807EaBX0fqt0E  
zUNxqQnDt3+c1uDkp20Jd8VTc7Rlibj+JtX1jlHhHr0m  
-----END CERTIFICATE-----
```

```
-----BEGIN PRIVATE KEY-----  
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwg5jAgEA  
yAfW6DBCdtvaSYqZknGa3W9Jqt7qsOXdsIroGmub5jzN  
TU0v1jnrxsSBiKa3ER/KVVkZ25xWsg5/dmBK/oBKXYsh  
TCMba6AK/UNsdlLZX60PjDk1B2J1psLRDAC/P+zRYa7h  
y2oQmWonJPHYE7V1jGd6CR1c2vSax6NAU89CygD+YkKc  
G5b0gTD5kdb+u1bbaQv7b+5TrZlwbe/wCfQ029uuzj4A  
/Aj/PrvrAgMBAECggEA0f0851mq05Lxw19oLXvvels  
wIni41GTD3d9k4M1r0EARtUHH9p+99AkxI0j14bILstL  
6Sx9wsHDXjFFjnYA2P3m9CQ35al8me2xD5HfgA8KACsy  
izX6G/p0V97uyG4nmEZmU59LCU1GJJqZ0pIybhdc2nJ  
ipI9/saPFDBcFyUDacv+CW595fp3r9Ahp3b3Fd0agrVL  
yTKXHZ1HgUjD3aK6TzHaM7s2fAPU82xxAq1rZhZycQKE  
43oTGpshrQYpBznNoS0dKkwvY9EABHydwGZexc8w3  
2PP6dJQC4eIA2NTUcNnfXZmZ6hRXjysQ5NUYg5RsHm21  
gfNpR/bgCnEeNdGvtQ9AbmVp0QKBgQC95Fk8zltSI4vK  
C/cRfe0mWlrkyNSGchqsHkPcKLQsnBeUIq7ID/9MS142  
p4EGh8UryB8LTtJnWCoMwyFY5QjtC7hjkkORC29Fdk  
KMTiGpM8+wKBgG0D560nBeVgTy+i0vxvUvRRQLz5jpRF  
Aixt17z+Q9yONYzeT8IhqNEDXpYbCcc1TsOXzVzme1uf  
pFYvAlu7dibJ18tuLWT8Wl0drTj1ZJN0789QfIb79qg4  
EEbcZ60cJbtFuCKxaya15pE5La/daVr1cfbA3Ue0TQ9y  
VsriVe61cHkHRWXB1FaXoF8cZgiCJTSrSiqGYVZAxoTC  
qvPuvni8o5i3076YzERCcIVm6rIp31kcIGIjCnEcgYEA  
m+kowoDAamZqh&TTUy5imuUwjejee2GrBtDdyJaJ9w  
JiahCZ/NBo0/I7pXcIJMYXRKJD8cePBRHtv8S1MtReB2  
unuaWGHGm/8HFMpNCREFN4=  
-----END PRIVATE KEY-----
```

Partly shown due to security issues of the keys

After we already get the key, then we need to Generate a CSR and Private Key for the web server we use “openssl req -newkey rsa:2048 -nodes -keyout qaziwa1.tk.key -out qaziwa1.tk.csr

```
root@ip-10-0-1-171:~# openssl req -newkey rsa:2048 -nodes -keyout qaziwal.tk.  
.tk.csr  
Generating a RSA private key  
.....+++++  
.....+++++  
writing new private key to 'qaziwal.tk.key'  
----  
You are about to be asked to enter information that will be incorporated  
into your certificate request.  
What you are about to enter is what is called a Distinguished Name or a DN.  
There are quite a few fields but you can leave some blank  
For some fields there will be a default value,  
If you enter '.', the field will be left blank.  
----  
Country Name (2 letter code) [AU]:SG  
State or Province Name (full name) [Some-State]:qaziwal  
Locality Name (eg, city) []:qaziwal  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:qaziwal  
Organizational Unit Name (eg, section) []:  
Common Name (e.g. server FQDN or YOUR name) []:qaziwal.tk  
Email Address []:wafimuhd5@gmail.com  
  
Please enter the following 'extra' attributes  
to be sent with your certificate request  
A challenge password []:  
An optional company name []:  
root@ip-10-0-1-171:~# cat qaziwal.tk.csr
```

Then we fill up the credentials shown above.

We then copy the private key and certificate that is given to us from the cloudflare into a file. We called it qaziwal.tk.key (for the private key from cloudflare) and qaziwal.tk.crt (for certificate key from cloudflare)

```
root@ip-10-0-1-171:/home/ubuntu# nano qaziwal.tk.key  
root@ip-10-0-1-171:/home/ubuntu# nano qaziwal.tk.crt
```

Once we are done adding the keys into the file, then we need to go to nano /etc/apache2/sites-available/000-default.conf and edit the configuration shown below

```

<VirtualHost *:443>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@gaziwal.tk
    ServerName gaziwal.tk
    ServerAlias www.gaziwa.tk
        DocumentRoot /var/www/html
    SSLEngine on
    SSLCertificateFile /home/ubuntu/qaziwal.tk.crt
    SSLCertificateKeyFile /home/ubuntu/qaziwal.tk.key
        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
    SecRuleEngine On
    SecRule ARGS:modsecparam "@contains test" "id:4321,deny,status:403,msg:'ModSecurity test rule has triggered'"
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
<VirtualHost *:80>
    ServerName gaziwal.tk
    Redirect permanent / https://gaziwal.tk/
</VirtualHost>

```

After editing it, we then do a sudo a2enmod ssl, which is to enable ssl and we do a service apache2 restart, to restart the apache server.

```

root@ip-10-0-1-171:/home/ubuntu# sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
root@ip-10-0-1-171:/home/ubuntu# service apache2 restart
root@ip-10-0-1-171:/home/ubuntu#

```

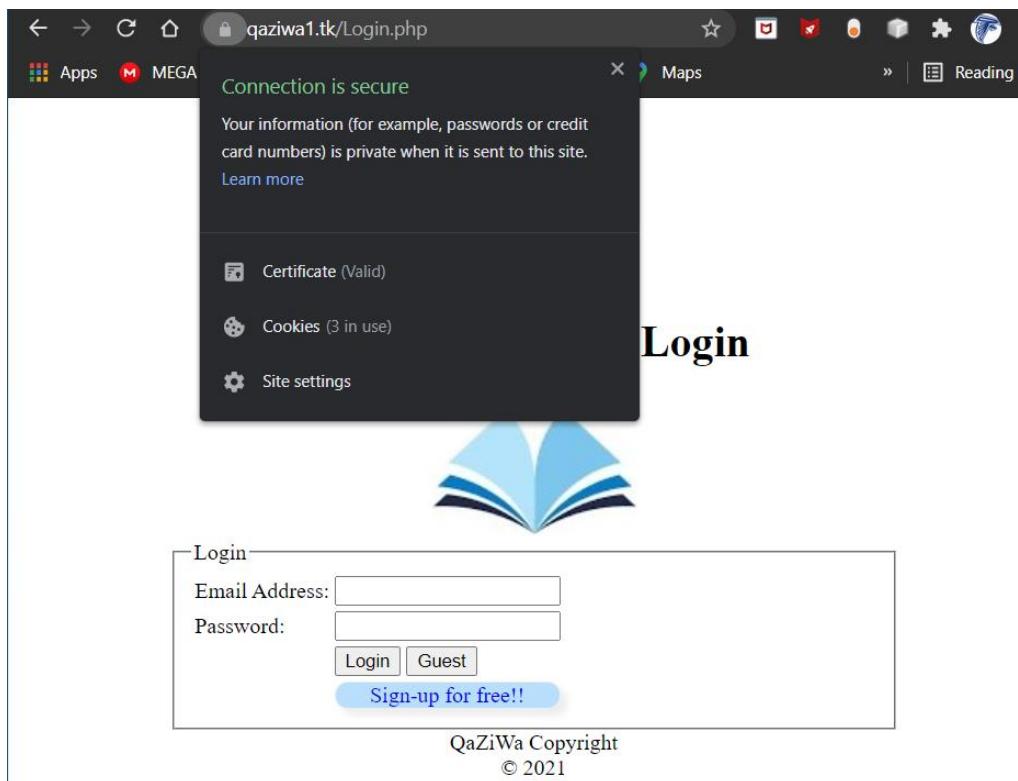
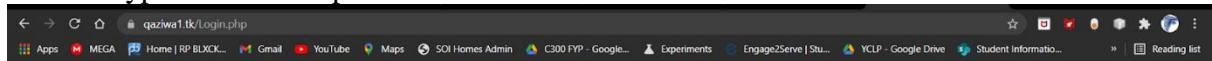
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zon
Danm Secure Qaziwa Website	i-01d0a821608a529fc	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a

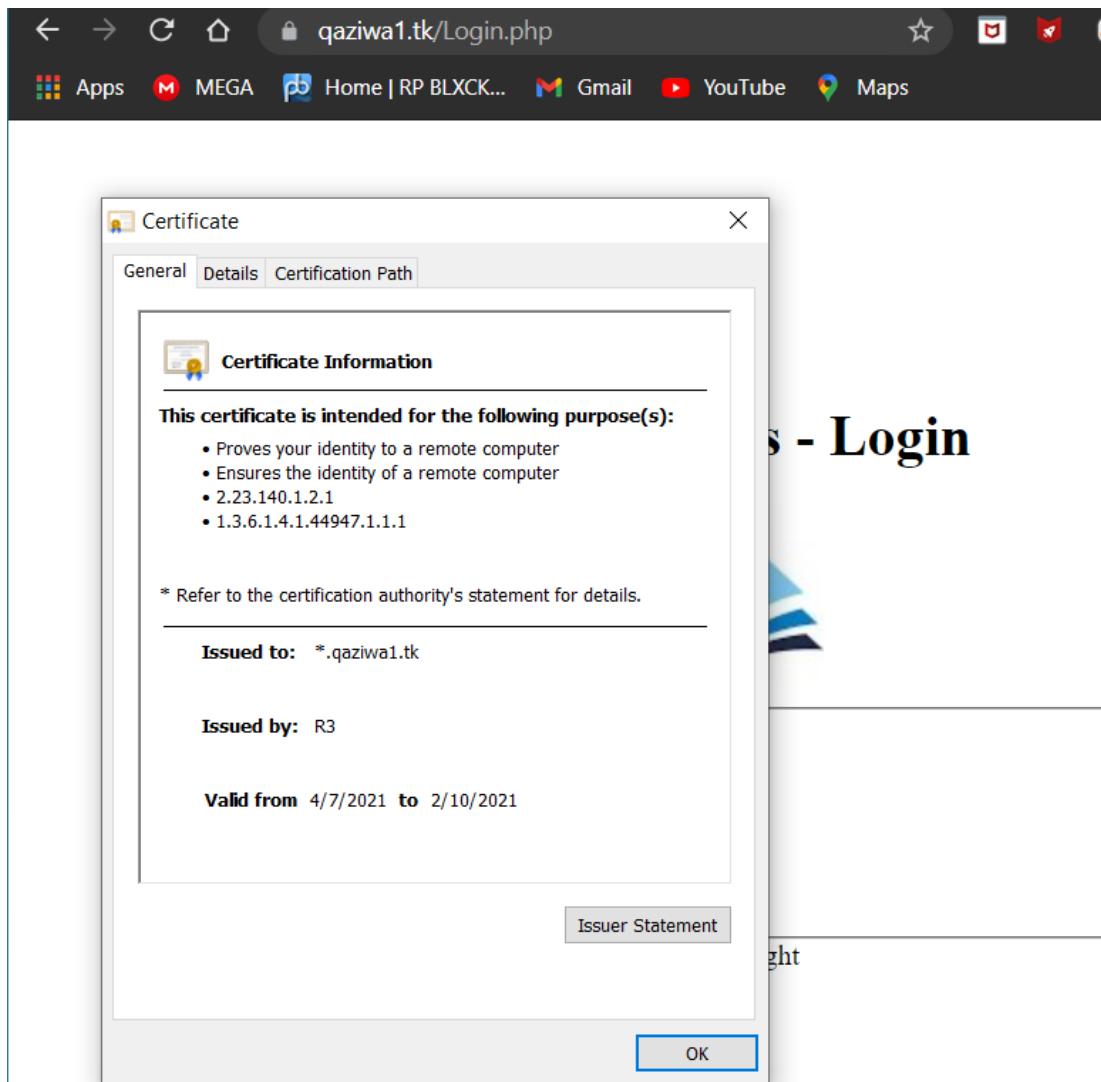
Inbound rules			
<input type="text"/> Filter rules			
Port range	Protocol	Source	Security groups
80	TCP	0.0.0.0/0	Qaziwa public security grp
22	TCP	0.0.0.0/0	Qaziwa public security grp
443	TCP	0.0.0.0/0	Qaziwa public security grp

Outbound rules			
Port range	Protocol	Destination	Actions
80	TCP	0.0.0.0/0	Qaziwa public security grp
22	TCP	0.0.0.0/0	Qaziwa public security grp
443	TCP	0.0.0.0/0	Qaziwa public security grp

After configuring that, we then need to add another rule which enables port 443 (HTTPS) due to the fact that we are enabling HTTPS in our website.

Then we wait for a while, to ensure the cert from the cloudflare is being applied to the web server. After a few minutes, we refresh the domain name (qaziwa1.tk) and see if there is an SSL encryption from the particular instance.





As shown above, we can see that a certificate is being applied to the web server and we want to check the cloudflare to see if the SSL/TLS is being set to Full (Strict) section.

Overview Edge Certificates Client Certificates Origin Server Custom Hostnames

✓ Your SSL/TLS encryption mode is Full (strict)

This setting was last changed 2 hours ago

Off (not secure) ⓘ
 No encryption applied

Flexible
 Encrypts traffic between the browser and Cloudflare

Full
 Encrypts end-to-end, using a self signed certificate on the server

Full (strict)
 Encrypts end-to-end, but requires a trusted CA or Cloudflare Origin CA certificate on the server

Learn more about [End-to-end encryption with Cloudflare](#)

[API ▶](#) [Help ▶](#)

Yes, it showed that the Full (strict) section is being applied from the client to the cloudflare and to the AWS ec2 instance. As such due to AWS not having enough privileges, of inserting a certificate manager and implementing it on AWS Educate, therefore, we can just implement a full SSL certificate on AWS EC2 instance.

The screenshot shows the AWS CloudWatch Metrics interface. At the top, there is a header with a checked checkbox, the text "Danm Secure Qaziwa Website", the ID "i-01d0a821608a529fc", a green "Running" status indicator, a magnifying glass icon, the instance type "t2.micro", and a green circular icon with "2/2 checks passed". Below the header, there is a large empty rectangular area. Underneath this area, the text "Instance: i-01d0a821608a529fc (Danm Secure Qaziwa Website)" is displayed. A navigation bar below it includes tabs for "Details" (which is selected), "Security", "Networking", "Storage", "Status checks", "Monitoring", and "Tags". The "Details" tab has a dropdown menu open under "Instance summary" with the option "Info" selected. A tooltip "Public IPv4 address copied" is shown over the "Info" button. The main content area displays the instance ID "i-01d0a821608a529fc (Danm Secure Qaziwa Website)", its public IP "52.6.166.103", and a link to "open address". To the right, it lists private IP addresses: "Private IPv4 addresses" and "10.0.1.171". Below this, the "Instance state" is shown as "Running" with a green arrow icon. The URL bar at the bottom of the browser window shows "52.6.166.103". The page content is a black Google search results page with the word "Google" prominently displayed.

Now instead we try to access the IP address directly from the instance and enter it on the google URL search.

The screenshot shows a web browser with the URL "qaziwa1.tk/Login.php" entered in the address bar. The browser's toolbar includes icons for Apps, MEGA, Home, Gmail, YouTube, Maps, SOI Homes Admin, C300 FYP - Google..., Experiments, Engage2Serve, and YCLP. The main content area displays a login form titled "QaZiWa Books - Login". The form contains fields for "Email Address" and "Password", and buttons for "Login" and "Guest". Below the form is a blue button labeled "Sign-up for free!!". At the bottom of the page, there is a copyright notice: "QaZiWa Copyright © 2021".

QaZiWa Books - Login



Login:

Email Address:

Password:

[Sign-up for free!!](#)

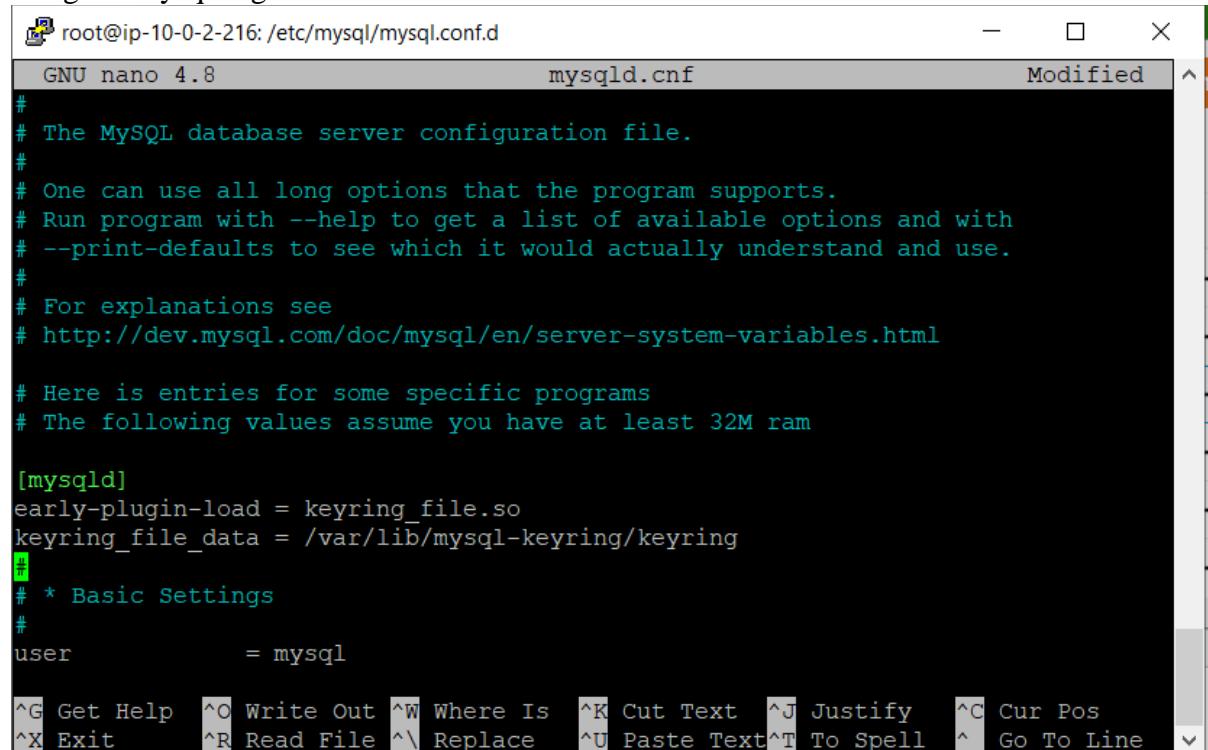
QaZiWa Copyright
© 2021

As you can see it will automatically change the IP address to qaziwa1.tk. From here we know that it is stable and the connection is secure. Since we already settled with encrypting data in transmission using HTTPs now is the time to encrypt data at rest, which is the database side.

Database Encryption:

Innodb encryption:

Encryption of the database is very important so that when the attacker wants to exfiltrate the database, it can ensure that all of the data the attacker is being encrypted and therefore, making sure that the attacker will not have the user data on his hands in plaintext. We will be using Innodb to encrypt our database and its table so that we can ensure total security in our database and its tables. As such, below is our screenshot on the encryption of the database using the mysql engine which is Innodb.



```
root@ip-10-0-2-216:/etc/mysql/mysql.conf.d# nano mysqld.cnf
GNU nano 4.8
mysqld.cnf
Modified

#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
early-plugin-load = keyring_file.so
keyring_file_data = /var/lib/mysql-keyring/keyring
#
# * Basic Settings
#
user          = mysql

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit     ^R Read File  ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line
```

Firstly we go to the directory “cd /etc/mysql/mysql.conf.d” and “nano mysqld.cnf”. After seeing the configuration page of the mysql, we can see, edit and add “early-plugin-load = keyring_file.so” and “keyring_file_data = /var/lib/mysql-keyring/keyring”, then we save the file.

.
After saving the file, we then enter the into the mysql section shown below

```
root@ip-10-0-2-216:/etc/mysql/mysql.conf.d# ls
mysql.cnf  mysqld.cnf
root@ip-10-0-2-216:/etc/mysql/mysql.conf.d# mysql -u qaziwa -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.25-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> [REDACTED]
```

Using “mysql -u qaziwa -p” to login into the database, and then after that we enter the required password to enter.

Upon entering we enter the following commands shown below to install and load a plugin into the database, configure the keyring file.

```
mysql> INSTALL PLUGIN keyring_file SONAME 'keyring_file.so';
Query OK, 0 rows affected (0.02 sec)

mysql> SET GLOBAL keyring_file_data = '/var/lib/mysql-keyring/keyring';
Query OK, 0 rows affected (0.00 sec)

mysql> [REDACTED]
```

So now, we can create encrypted tables and even alter existing tables to encrypt them.

Below is the list of encryption that we use on the database level and the tables level.

```
mysql> ALTER SCHEMA qaziwa DEFAULT ENCRYPTION = 'Y';
Query OK, 1 row affected (0.01 sec)

mysql> [REDACTED]
```

On the above, it is a database level encryption that we encrypt the database named ““qaziwa””

```
mysql> use qaziwa;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_qaziwa |
+-----+
| books           |
| cart_books      |
| users           |
+-----+
3 rows in set (0.01 sec)

mysql> [REDACTED]
```

After encrypting the database level, then we go deep to encrypt the tables in the database. So we view which tables are in the Qaziwa database.

```
mysql> ALTER TABLE users ENCRYPTION='Y';
Query OK, 7 rows affected (0.06 sec)
Records: 7  Duplicates: 0  Warnings: 0

mysql> ALTER TABLES books ENCRYPTION='Y';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near 'TABLES
books ENCRYPTION='Y'' at line 1
mysql> ALTER TABLE books ENCRYPTION='Y';
Query OK, 11 rows affected (0.06 sec)
Records: 11  Duplicates: 0  Warnings: 0

mysql> ALTER TABLW cart_books ENCRYPTION='Y';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near 'TABLW
cart_books ENCRYPTION='Y'' at line 1
mysql> ALTER TABLE cart_books ENCRYPTION='Y';
Query OK, 8 rows affected (0.07 sec)
Records: 8  Duplicates: 0  Warnings: 0

mysql>
```

After we know the tables that are inside the Qaziwa database, we now proceed to encrypt that to add a second layer of defense. As shown in a highlighted above, we encrypt the users, books, cart_books. This is to ensure that the tables cannot be seen in plaintext when the attacker manages to dump out the database.

Going further, we will be using a Master Key Rotation, this means that the master encryption key will be periodically rotated and let say the key is compromised, the attack will not have the suitable key to decrypt the tables and database, due to the key will keep changing. Everytime the master key is rotated, all tablespace keys in the MYSQL instance will be re-encrypted and saved back to their own respective tablespace headers. The rotating master encryption key will only change the master encryption key and re-encrypt the tablespace keys. It does not decrypt or re-encrypt the tablespace data.

> Table space is a storage location where the actual data in the database is stored.

```
mysql> ALTER INSTANCE ROTATE INNODB MASTER KEY;
Query OK, 0 rows affected (0.00 sec)
```

So to enable the Master Key Rotation, we will be using the command “ALTER INSTANCE ROTATE INNODB MASTER KEY;”

After finishing the database and table encryption command using innodb engine, now we need to verify if the database and tables are encrypted or not.

```

mysql> #Now time to verify;
mysql> SELECT TABLE_SCHEMA, TABLE_NAME, CREATE_OPTIONS FROM
    ->     INFORMATION_SCHEMA.TABLES WHERE CREATE_OPTIONS LIKE '%ENCRYPTION%';
+-----+-----+-----+
| TABLE_SCHEMA | TABLE_NAME | CREATE_OPTIONS |
+-----+-----+-----+
| qaziwa      | books      | ENCRYPTION='Y' |
| qaziwa      | cart_books | ENCRYPTION='Y' |
| qaziwa      | users      | ENCRYPTION='Y' |
+-----+-----+-----+
3 rows in set (0.01 sec)

mysql> █

```

In the above section, we can see that the tables in the database are encrypted.

```

mysql> SELECT SCHEMA_NAME, DEFAULT_ENCRYPTION FROM
    -> INFORMATION_SCHEMA.SCHEMATA WHERE DEFAULT_ENCRYPTION='YES';
+-----+-----+
| SCHEMA_NAME | DEFAULT_ENCRYPTION |
+-----+-----+
| qaziwa      | YES                 |
+-----+-----+
1 row in set (0.00 sec)

mysql> █

```

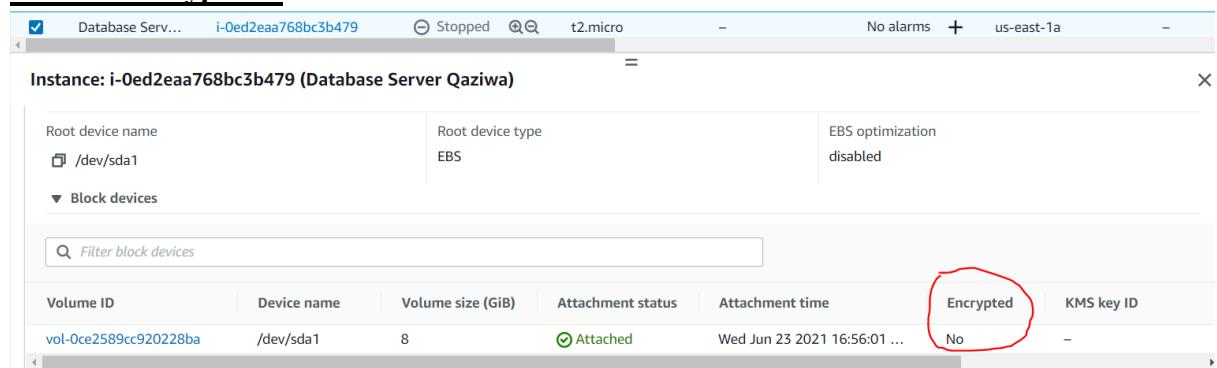
In the above section, we can see that the database is encrypted.

Therefore, encryption in the database is important to prevent any attackers from exfiltrating the database and the tables and put it into an advantage and gain the user information. Hence, the database encryption, we can certainly make sure that all of the information in the database will not be seen in plaintext.

Going further than this, it is also recommended also to encrypt the Database Instance via the EBS volume of the instance. EBS volume is durable, which is a block-level storage device that we can attach on the instance. So, as such we need to encrypt the EBS volume so that to ensure that stored data is secure. This is also part of defense in depth in which we encrypt the Database volume and ensure that when an attacker gets the instance volume, it would not be in readable plaintext but instead an encrypted volume using the AWS Key Management.

As shown below, it is shown that the Database instance volume is not encrypted.

Volume encryption:



Now we know that it is not encrypted, now is the time to encrypt the volume of the database. To start off, we need to create a symmetric key from AWS Key Management Service (KMS) to ensure that we can use the key to encrypt the EBS volume.

We do the configuration shown below.

The screenshot shows the 'Key type' configuration screen. It has two main sections: 'Key type' and 'Advanced options'.

Key type (Help me choose [?](#))

- Symmetric**
A single encryption key that is used for both encrypt and decrypt operations
- Asymmetric**
A public and private key pair that can be used for encrypt/decrypt or sign/verify operations

Advanced options

Key material origin (Help me choose [?](#))

- KMS**
- External
- Custom key store (CloudHSM)

Regionality (You cannot change this setting after the key is created. Help me choose [?](#))

- Single-Region key**
Never allow this key to be replicated into other Regions
- Multi-Region key**
Allow this key to be replicated into other Regions

Cancel **Next**

Then we click next

Add labels

Alias

You can change the alias at any time. [Learn more](#)

Alias

Qaziwa-EBSEncryption

Description - optional

You can change the description at any time.

Description - optional

This is an encryption key for EBS Volume



Tags - optional

You can use tags to categorize and identify your CMKs and help you track your AWS costs. When you add tags to AWS resources, AWS generates a cost allocation report for each tag. [Learn more](#)

This key has no tags.

[Add tag](#)

You can add up to 50 more tags.

[Cancel](#)

[Previous](#)

[Next](#)

We filled up the section above and click on next

<input type="checkbox"/>	AWSServiceRoleForElastiCache	/aws-service-role/elasticache.amazonaws.com/	Role
<input type="checkbox"/>	AWSServiceRoleForElasticLoadBalancing	/aws-service-role/elasticloadbalancing.amazonaws.com/	Role
<input type="checkbox"/>	AWSServiceRoleForOrganizations	/aws-service-role/organizations.amazonaws.com/	Role

Key deletion

Allow key administrators to delete this key.

[Cancel](#)

[Previous](#)

[Next](#)

The screenshot shows the 'Add Key' wizard in the AWS KMS console. Step 3: Set permissions for this key. It lists three service roles:

Role	ARN
AWSServiceRoleForElasticCache	role/elasticache.amazonaws.com/
AWSServiceRoleForElasticLoadBalancing	/aws-service-role/elasticloadbalancing.amazonaws.com/
AWSServiceRoleForOrganizations	/aws-service-role/organizations.amazonaws.com/

Below this is the 'Other AWS accounts' section, which contains the following text:

Specify the AWS accounts that can use this key. Administrators of the accounts you specify are responsible for managing the permissions that allow their IAM users and roles to use this key. [Learn more](#)

[Add another AWS account](#)

At the bottom are 'Cancel', 'Previous', and 'Next' buttons.

From the above, we must not select anything and click on next twice.

Then from here you can see the review section shown below.

The screenshot shows the 'Review' section of the 'Add Key' wizard. It displays the key configuration and alias/description details.

Key configuration

Key type Symmetric	Key spec SYMMETRIC_DEFAULT	Key usage Encrypt and decrypt
Origin AWS_KMS	Regionality Single-Region key	

i You cannot change the key configuration after the key is created.

Alias and description

Alias Qaziwa-EBSEncryption	Description This is an encryption key for EBS Volume
-------------------------------	---

Tags

Key	Value
No data	
No tags to display	

Key policy

To change this policy, return to previous steps or edit the text here.

```

1 {
2   "Id": "key-consolepolicy-3",
3   "Version": "2012-10-17",
4   "Statement": [
5     {
6       "Sid": "Enable IAM User Permissions",
7       "Effect": "Allow",
8       "Principal": {
9         "AWS": "arn:aws:iam::501144450828:root"
10      },
11      "Action": "kms:*",
12      "Resource": "*"
13    }
14  ]
15 }
```

Cancel Previous **Finish**

Then click on finish.

Then in the key policy, we add administrator inside

Key policy

Key administrators

Choose the IAM users and roles who can administer this key through the KMS API. You might need to add additional permissions for the users or roles to administer this key from this console. [Learn more](#)

Add	Remove	
<input type="button"/>	<input type="button"/>	
< 1 >		
<input type="checkbox"/> Name	Path	Type
<input type="checkbox"/> qaziwa-admin	/	User

Key deletion

Allow key administrators to delete this key

Key users

The following IAM users and roles can use this key for cryptographic operations. They can also allow AWS services that are integrated with KMS to use the key on their behalf. [Learn more](#)

Add	Remove	
<input type="button"/>	<input type="button"/>	
< 1 >		
<input type="checkbox"/> Name	Path	Type
<input type="checkbox"/> qaziwa-admin	/	User

Other AWS accounts

Once that is done, now we need to go to the EBS Volume section shown below

Amazon EBS Volume Details										
Actions		Name	Volume ID	Size	Volume Type	IOPS	Throughput	Snapshot	Created	Availability Zone
vol-0ce2589cc920228ba	8 GiB	gp2	100	-	snap-0557dea...	June 23, 2021 at 4:...	us-east-1a	in-use	None	

Now we need to create a snapshot for this, by doing on Action > Create Snapshot

Volume vol-0ce2589cc920228ba i

Description Database-EBS Encryption i

Encrypted Not Encrypted i

Key (128 characters maximum)	Value (256 characters maximum)
-------------------------------------	---------------------------------------

This resource currently has no tags

Choose the Add tag button or [click to add a Name tag](#)

Add Tag
50 remaining (Up to 50 tags maximum)

Cancel
Create Snapshot

And then click on create snapshot.

After this go to the snapshot section, and locate the snapshot that we have done and wait for a while until it is fully created.

Create Snapshot						Actions	
Owned By Me		Filter by tags and attributes or search by keyword				1 to 14 of 14	
	Name	Snapshot ID	Size	Description			
		snap-0219018441ct...	8 GiB	Created by CreateImage(i-0eec8b9304eb2e77e) for ami-0...			
		snap-0447d4a9a50...	8 GiB	Created by CreateImage(i-0eec8b9304eb2e77e) for ami-0...			
	Database Vo...	snap-0557dea4013...	8 GiB	Created on 2/6/2021			
		snap-08472064cd6...	8 GiB	Created by CreateImage(i-0ed2eaa768bc3b479) for ami-0...			
		snap-093cf8765fe9...	8 GiB	Created by CreateImage(i-0eec8b9304eb2e77e) for ami-0...			
		snap-0a7f41164979...	8 GiB	Created by CreateImage(i-0eec8b9304eb2e77e) for ami-0...			
		snap-0b7ef6c93ba4...	8 GiB	Database-EBS Encryption			
		snap-0ca6666bddd...	8 GiB	Created by CreateImage(i-0eec8b9304eb2e77e) for ami-0...			
	Web Server ...	snap-0d758b25fa0c...	8 GiB	Snapshot on 16/6/2021			

Once the creation of the Database-EBS Encryption, Now we need to select it > action > create volume. Fill up the configuration that is shown below.

Create Volume

Snapshot ID: snap-0b7ef6c93ba4271ed

Volume Type: General Purpose SSD (gp2)

Size (GiB): 8 (Min: 1 GiB, Max: 16384 GiB)

IOPS: 100 / 3000 (Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS)

Throughput (MB/s): Not applicable

Availability Zone*: us-east-1a

Fast Snapshot Restore: Not enabled

Encryption: Encrypt this volume

Master Key: Qaziwa-EBSEncryption

KMS Key Description: This is an encryption key for EBS Volume

KMS Key Account: This account (501144450828)

KMS Key ID: ff2a79d0-adf2-4434-8ce8-46004e5079b1

KMS Key ARN: arn:aws:kms:us-east-1:501144450828:key/ff2a79d0-adf2-4434-8ce8-46004e5079b1

Volumes that are created from encrypted snapshots are automatically encrypted, and volumes that are created from unencrypted snapshots are automatically unencrypted. If no snapshot is selected, you can choose to encrypt the volume and specify your own key. Learn more about KMS keys

Key (128 characters maximum)	Value (256 characters maximum)
Name	Database-Encrypted EBS
Add Tag 49 remaining (Up to 50 tags maximum)	

* Required

Cancel Create Volume

After this create volume

Moving on, we go back to the EBS volume section and locate the “Database-Encrypted EBS”

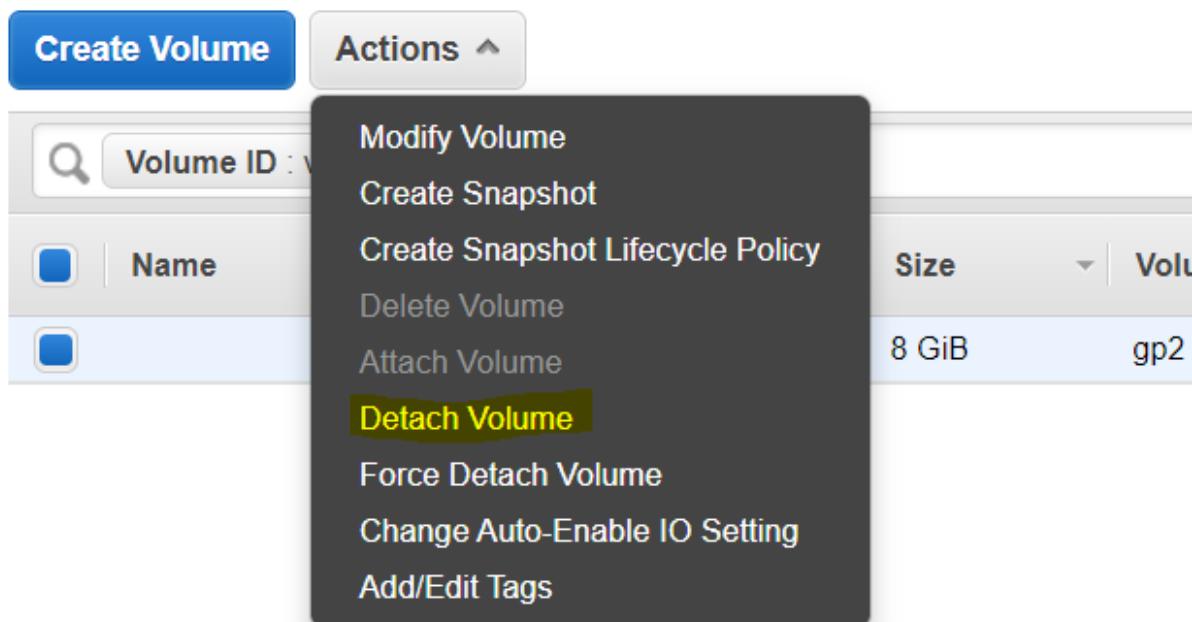
Amazon EBS Volume List											
Name	Volume ID	Size	Volume Type	IOPS	Throughput	Snapshot	Created	Availability Zone	State	Alarm Status	Attachments
Database-Encrypted EBS	vol-043d3cd...	8 GiB	gp2	100	-	snap-0b7ef8c9...	July 18, 2021 at 12:...	us-east-1a	available	None	
Test Database Server	vol-0ab2e5...	8 GiB	gp2	100	-	snap-0847206...	July 16, 2021 at 11:...	us-east-1a	available	None	

Actions | Monitoring | Volume Status | Encryption | KMS Key ID | KMS Key Alias | Multi-Attach Enabled

Okay Encrypted ff2a79d0-adf2... Qaziwa-EBSE... No

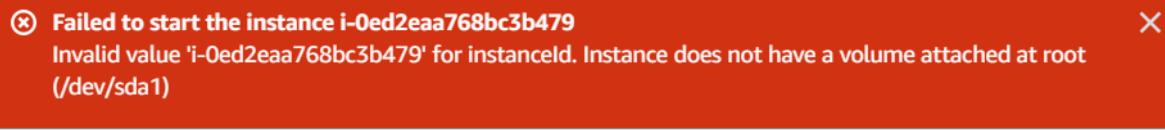
As shown above, we have successfully created the EBS volume that is encrypted.

After this, we need to detach the current non-encrypted EBS Volume that is attached in the Qaziwa Database Server. So to do this, we go back to the instance, click on the Database server and after that go to storage and click on the volume ID to locate the volume that is attached to the database server. Then we select the Volume ID > click on actions > and click on detach volume > click on confirm detach. (Side note: Make sure that the instance is being in stopped position)

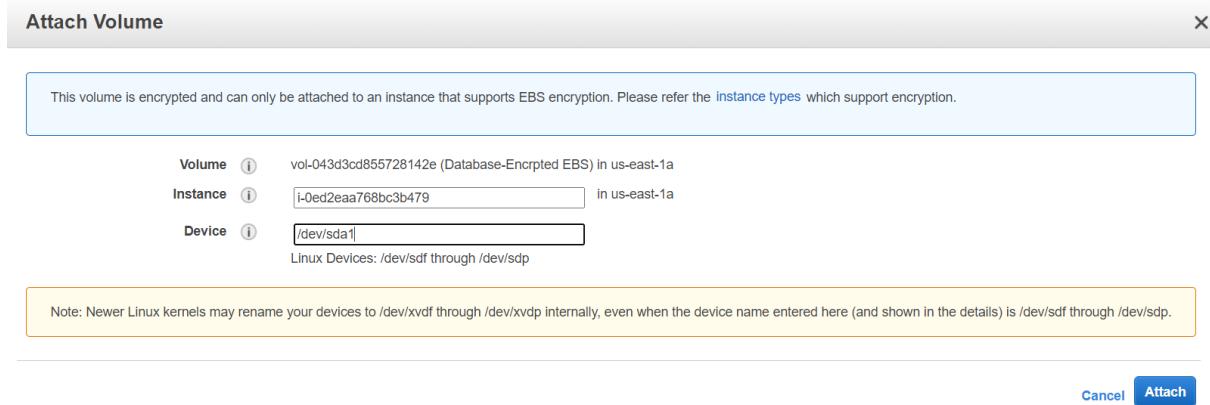


Once detach, then we need to wait for a while after it has fully detached from the database instance server.

Once done, go back to the database instance, and capture the error message,



So now we need to attach the volume that we have created with the device name /dev/sda1. To do this, we go back to the volume section and locate the "Database-Encrypted EBS" volume. We click on it and click on action > click on attach volume > select the Qaziwa Database Instance and for the device, choose /dev/sda1 shown below.



Then click on attach.

Once that is done, go back to the instance, and locate the Qaziwa Database Server and goto storage and see if the volume is already attached to the instance.

Volume ID	Device name	Volume size (GiB)	Attachment status	Attachment time	Encrypted	KMS key ID	Delete on Termination
vol-043d3cd855728142e	/dev/sda1	8	Attached	Sun Jul 18 2021 12:41:21 G...	Yes	ff2a79d0-adf2-4434-8ce8-46004e5079...	No

As shown above, we can see that we have successfully attached an encrypted volume into the instance and now is the time to test. We start the instance and see if it is started.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Pul
Database Server Qaziwa	i-0ed2eaa768bc3b479	Running	t2.micro	Initializing	No alarms	us-east-1a	-

Now it is running, we have successfully detach the encrypted EBS volume and attach an encrypted EBS volume.

Creation of Monitoring system, Privileges and Encryption in AWS:

It is important to implement a monitoring system in the project and in our service as it can help to take note of the traffic, flag out any alerts with a specific rule and much more. As

such help could help us in ensuring what time there might be heavy traffic, the CPU utilization, creation, traffic and much more.

Creation of KMS:

KMS is a centralized zone for storing all cryptographic keys to protect data. It is integrated with CloudTrail for monitoring of the key usage from the logs to allocate required resources across the network.

Despite having insufficient privileges, we will proceed and create KMS to show the rough idea of how the structure will be.

All Users Overview:

The screenshot shows the AWS KMS 'Customer managed keys' list page. At the top, there is a breadcrumb navigation: 'KMS > Customer managed keys'. Below the breadcrumb is a search bar labeled 'Filter keys by properties or tags'. On the right side of the header, there are buttons for 'Key actions' (with a dropdown arrow) and 'Create key' (in orange). The main area displays a table titled 'Customer managed keys (4)'. The table has columns: 'Aliases' (with a dropdown arrow), 'Key ID', 'Status', 'Key spec', and 'Key usage'. The table contains the following data:

Aliases	Key ID	Status	Key spec	Key usage
Qaziwa-AdminKey	f23afe98-9689-4274-b2b0-9c939a41f956	Enabled	RSA_2048	Encrypt and decrypt
QaziwaSymmetricKey	f320f17a-9a0c-418f-be2f-e4d54bdf562a	Pending deletion	SYMMETRIC_DEFAULT	Encrypt and decrypt
Qaziwa-NormalUser	ef7b5354-3288-4bf8-a037-25b468ce7819	Enabled	RSA_2048	Encrypt and decrypt
Qaziwa-DeveloperKey	44bbb11a-1205-4f3f-95a8-15fa75d8dfb0	Enabled	RSA_2048	Encrypt and decrypt

We will be creating 3 KMS keys for the different users. The first key called the “adminkey” will be allocated for the administrator account of the infrastructure. They have the most privileges for being able to control all users within the system. The second key called “normaluser” will be allocated for users accessing the website. Only a small amount of privileges should be allocated as they should not be managing the system. The third key called the “developerkey” will be allocated for the designers and engineers. The privileges allocated should be between admin and normal users.

Admin Key:

An asymmetric key is assigned to allow only the receiver which is the admin to decrypt the message using their private key. Key usage will be solely for encrypting and decrypting with the use of the allocated keys. It will be a single-region key used within the system.

KMS > Customer managed keys > Create key

Step 1
Configure key

Step 2
Add labels

Step 3
Define key administrative permissions

Step 4
Define key usage permissions

Step 5
Review

Configure key

Key type [Help me choose](#)

Symmetric
A single encryption key that is used for both encrypt and decrypt operations

Asymmetric
A public and private key pair that can be used for encrypt/decrypt or sign/verify operations

Key usage [Help me choose](#)

Encrypt and decrypt
Key pairs for public key encryption
Uses the public key for encryption and the private key for decryption.

Sign and verify
Key pairs for digital signing
Uses the private key for signing and the public key for verification.

Key spec [Help me choose](#)

RSA_2048

RSA_3072

RSA_4096

Advanced options

Regionality
You cannot change this setting after the key is created. [Help me choose](#)

Single-Region key
Never allow this key to be replicated into other Regions

Multi-Region key
Allow this key to be replicated into other Regions

[Cancel](#) [Next](#)

Admin has the most privilege. Therefore, they can delete when required.

KMS > Customer managed keys > Create key

Step 1
Configure key

Step 2
Add labels

Step 3
Define key administrative permissions

Step 4
Define key usage permissions

Step 5
Review

Add labels

Alias
You can change the alias at any time. [Learn more](#)

Alias
Qaziwa-AdminKey

Description - optional
You can change the description at any time.

Description - optional
Admin Key

Tags - optional

You can use tags to categorize and identify your CMKs and help you track your AWS costs. When you add tags to AWS resources, AWS generates a cost allocation report for each tag. [Learn more](#)

Tag key	Tag value - optional
<input type="text" value="Name"/> X	<input type="text" value="Qaziwa-AdminKey"/> X
Add tag	Delete tag
Enter a tag value	

You can add up to 49 more tags.

Cancel
Previous
Next

Configure key

Step 2 [Add labels](#)

Step 3 **Define key administrative permissions**

Step 4 [Define key usage permissions](#)

Step 5 [Review](#)

Key administrators

Choose the IAM users and roles who can administer this key through the KMS API. You may need to add additional permissions for the users or roles to administer this key from this console. [Learn more](#)

<input type="checkbox"/>	Name	Path	Type
<input checked="" type="checkbox"/>	qaziwa-admin	/	User
<input type="checkbox"/>	Qaziwa-Developer	/	User
<input type="checkbox"/>	Qaziwa-NormalUser	/	User
<input type="checkbox"/>	AWSServiceRoleForAmazonGuardDuty	/aws-service-role/guardduty.amazonaws.com/	Role
<input type="checkbox"/>	AWSServiceRoleForAutoScaling	/aws-service-role/autoscaling.amazonaws.com/	Role
<input type="checkbox"/>	AWSServiceRoleForAWSCloud9	/aws-service-role/cloud9.amazonaws.com/	Role

Key deletion

Allow key administrators to delete this key.

Cancel
Previous
Next

Since this key is allocated for administrator users, the admin is selected for the IAM role.

Define key usage permissions

This account			
Select the IAM users and roles that can use the CMK in cryptographic operations. Learn more			
	Name	Path	Type
<input checked="" type="checkbox"/>	qaziwa-admin	/	User
<input type="checkbox"/>	Qaziwa-Developer	/	User
<input type="checkbox"/>	Qaziwa-NormalUser	/	User
<input type="checkbox"/>	AWSServiceRoleForAmazonGuardDuty	/aws-service-role/guardduty.amazonaws.com/	Role
<input type="checkbox"/>	AWSServiceRoleForAutoScaling	/aws-service-role/autoscaling.amazonaws.com/	Role
<input type="checkbox"/>	AWSServiceRoleForAWSCloud9	/aws-service-role/cloud9.amazonaws.com/	Role
<input type="checkbox"/>	AWSServiceRoleForCloudWatchEvents	/aws-service-role/events.amazonaws.com/	Role

Review

Key configuration		
Key type Asymmetric	Key spec RSA_2048	Key usage Encrypt and decrypt
Origin AWS_KMS	Regionality Single-Region key	
<p>ⓘ You cannot change the key configuration after the key is created.</p>		

Alias and description	
Alias Qaziwa-AdminKey	Description Admin Key

Tags	
Key	Value
Name:	Qaziwa-AdminKey

Key policy	
To change this policy, return to previous steps or edit the text here.	
<pre>44 "Action": [45 "kms:Encrypt", 46 "kms:Decrypt", 47 "kms:ReEncrypt*", 48 "kms:DescribeKey", 49 "kms:GetPublicKey" 50], 51 "Resource": "*" 52 }, 53 { 54 "Sid": "Allow attachment of persistent resources", 55 "Effect": "Allow", </pre>	

Customer Key:

The symmetric key is used easier for encryption and decryption to the normal users. Keys are in a single region to prevent duplicated keys from lying around different regions.

Configure key

Key type [Help me choose](#)

Symmetric
A single encryption key that is used for both encrypt and decrypt operations

Asymmetric
A public and private key pair that can be used for encrypt/decrypt or sign/verify operations

▼ Advanced options

Key material origin
[Help me choose](#)

KMS

External

Custom key store (CloudHSM)

Regionality
You cannot change this setting after the key is created. [Help me choose](#)

Single-Region key
Never allow this key to be replicated into other Regions

Multi-Region key
Allow this key to be replicated into other Regions

[Cancel](#) [Next](#)

Description for the normal user.

Add labels

Alias
You can change the alias at any time. [Learn more](#)

Alias
Qaziwa-NormalUser

Description - optional
You can change the description at any time.

Description - optional
Key for Normal User 

Tags - optional

You can use tags to categorize and identify your CMKs and help you track your AWS costs. When you add tags to AWS resources, AWS generates a cost allocation report for each tag. [Learn more](#)

Tag key Enter a tag value Delete tag

You can add up to 49 more tags.

[Cancel](#) [Previous](#) [Next](#)

Normal users should not have the privilege to get hold of the key, hence admin and developer are allocated to control the key.

Define key administrative permissions

Key administrators

Choose the IAM users and roles who can administer this key through the KMS API. You may need to add additional permissions for the users or roles to administer this key from this console. [Learn more](#)

Name	Path	Type
<input checked="" type="checkbox"/> qaziwa-admin	/	User
<input checked="" type="checkbox"/> Qaziwa-Developer	/	User
<input type="checkbox"/> Qaziwa-NormalUser	/	User
<input type="checkbox"/> AWSServiceRoleForAmazonGuardDuty	/aws-service-role/guardduty.amazonaws.com/	Role
<input type="checkbox"/> AWSServiceRoleForAutoScaling	/aws-service-role/autoscaling.amazonaws.com/	Role

Key deletion

Allow key administrators to delete this key.

Cancel Previous **Next**

Since this key is allocated for customer/ normal users, the normal user is selected for the IAM role.

Define key usage permissions

This account

Select the IAM users and roles that can use the CMK in cryptographic operations. [Learn more](#)

Name	Path	Type
<input type="checkbox"/> qaziwa-admin	/	User
<input type="checkbox"/> Qaziwa-Developer	/	User
<input checked="" type="checkbox"/> Qaziwa-NormalUser	/	User
<input type="checkbox"/> AWSServiceRoleForAmazonGuardDuty	/aws-service-role/guardduty.amazonaws.com/	Role
<input type="checkbox"/> AWSServiceRoleForAutoScaling	/aws-service-role/autoscaling.amazonaws.com/	Role
<input type="checkbox"/> AWSServiceRoleForAWSCloud9	/aws-service-role/cloud9.amazonaws.com/	Role

Review

Key configuration

Key type Asymmetric	Key spec RSA_2048	Key usage Encrypt and decrypt
Origin AWS_KMS	Regionality Single-Region key	

i You cannot change the key configuration after the key is created.

Alias and description

Alias Qaziwa-NormalUser	Description Key for Normal User
----------------------------	------------------------------------

Tags

Key	Value
Name:	Qaziwa-NormalUser

Key policy

To change this policy, return to previous steps or edit the text here.

1 {

Developer Key:

An asymmetric key is assigned to allow only the receiver which is the developer to decrypt the message using their private key. Key usage will be solely for encrypting and decrypting with the use of the allocated keys. RSA2048 is enough for now. It will be a single-region key used within the system.

Configure key

Key type Help me choose

Symmetric

A single encryption key that is used for both encrypt and decrypt operations

Asymmetric

A public and private key pair that can be used for encrypt/decrypt or sign/verify operations

Key usage Help me choose

Encrypt and decrypt

Key pairs for public key encryption

Uses the public key for encryption and the private key for decryption.

Sign and verify

Key pairs for digital signing

Uses the private key for signing and the public key for verification.

Key spec [Help me choose](#)

RSA_2048
 RSA_3072
 RSA_4096

▼ Advanced options

Regionality
You cannot change this setting after the key is created. [Help me choose](#)

Single-Region key
Never allow this key to be replicated into other Regions

Multi-Region key
Allow this key to be replicated into other Regions

[Cancel](#) [Next](#)

Description of developer key.

Add labels

Alias
You can change the alias at any time. [Learn more](#)

Alias
Qaziwa-DeveloperKey

Description - optional
You can change the description at any time.

Description - optional
A Key for Developer

Tags - optional

You can use tags to categorize and identify your CMKs and help you track your AWS costs. When you add tags to AWS resources, AWS generates a cost allocation report for each tag. [Learn more](#)

Tag key Enter a tag value Delete tag

You can add up to 49 more tags.

[Cancel](#) [Previous](#) [Next](#)

Since admin has the most privilege, the developer should not have the privilege to get hold of the key letting only the admin control the key.

Define key administrative permissions

Key administrators

Choose the IAM users and roles who can administer this key through the KMS API. You may need to add additional permissions for the users or roles to administer this key from this console. [Learn more](#)

	Name	Path	Type
<input checked="" type="checkbox"/>	qaziwa-admin	/	User
<input type="checkbox"/>	Qaziwa-Developer	/	User
<input type="checkbox"/>	Qaziwa-NormalUser	/	User
<input type="checkbox"/>	AWSServiceRoleForAmazonGuardDuty	/aws-service-role/guardduty.amazonaws.com/	Role
<input type="checkbox"/>	AWSServiceRoleForAutoScalina	/aws-service-	Role

Key deletion

Allow key administrators to delete this key.

Cancel [Previous](#) **Next**

Since this key is allocated for customer/ normal users, the normal user is selected for the IAM role.

Define key usage permissions

This account

Select the IAM users and roles that can use the CMK in cryptographic operations. [Learn more](#)

	Name	Path	Type
<input type="checkbox"/>	qaziwa-admin	/	User
<input checked="" type="checkbox"/>	Qaziwa-Developer	/	User
<input type="checkbox"/>	Qaziwa-NormalUser	/	User
<input type="checkbox"/>	AWSServiceRoleForAmazonGuardDuty	/aws-service-role/guardduty.amazonaws.com/	Role

Review

Key configuration

Key type Asymmetric	Key spec RSA_2048	Key usage Encrypt and decrypt
Origin AWS_KMS	Regionality Single-Region key	

Info You cannot change the key configuration after the key is created.

Alias and description

Alias Qaziwa-DeveloperKey	Description A Key for Developer
------------------------------	------------------------------------

The screenshot shows the AWS IAM Key Management interface. At the top, there's a table for 'Tags' with one entry: 'Name: Qaziwa-DeveloperKey'. Below this is a section titled 'Key policy' containing the following JSON policy code:

```

6     "Sid": "Enable IAM User Permissions",
7     "Effect": "Allow",
8     "Principal": {
9         "AWS": "arn:aws:iam::501144450828:root"
10    },
11    "Action": "kms:*",
12    "Resource": "*"
13},
14{
15    "Sid": "Allow access for Key Administrators",
16    "Effect": "Allow",
17    "Principal": {
18        "AWS": "arn:aws:iam::501144450828:user/qaziwa-admin"
19    }

```

Since we finished our KMS configuration of the keys, now is to configure the CloudWatch.

Creation of CloudWatch:

AWS cloud watch will collect the monitoring and operational data by observing all the services used within the AWS classroom in the form of metrics and logs. To achieve this, we will create a dashboard called “QaziwaWebServer” to store all collected data from our cloud resource usage. The various widgets from the line graph, bar graph and even pie chart allow us to monitor the status differently. Alarms can also be set to trigger an alert when any abnormality is found, making troubleshooting easier.

The screenshot shows the AWS CloudWatch Metrics dashboard creation interface. On the left, a sidebar lists navigation options: Dashboards, Alarms (In alarm: 0, Insufficient data: 7, OK: 0), Logs (Log groups, Insights), Metrics (Explorer, Streams: New), and Events. The main area is titled 'Dashboards' and contains a 'Create dashboard' button. A table shows the details for the 'QaziwaWebServer' dashboard, which was last updated on 2021-06-30 07:25. The table columns are Name, Favorite, Share, and Last updated (UTC).

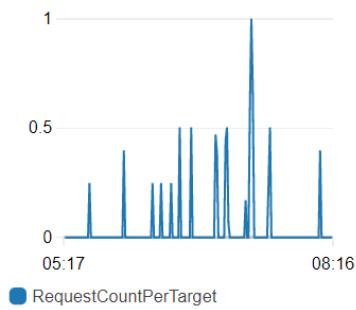
Name	Favorite	Share	Last updated (UTC)
QaziwaWebServer	☆	🔗	2021-06-30 07:25

In our scenario, the cloud watch will be monitored in 4 sections - load balancer, database server, web server, and auto scaling. For this section, the 3 graphs below monitor the load balancer. The first graph represents the request count per target. When there are users using Qaziwa's website, the graph will increase accordingly to the demand as shown below. If there are no users, the graph will remain at 0. The second graph consists of new connection, request and active connection. The third graph is the consumption of the load balancing capacity unit (LCU).

Load Balancer (Qaziwa)

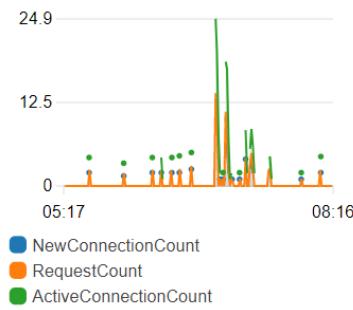
QaziwaLB_RequestCount...

RequestCountPerTarget



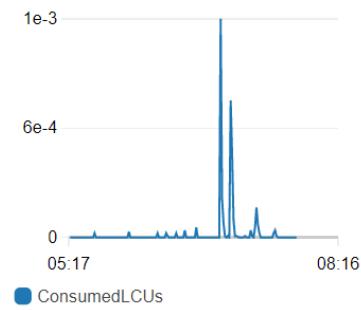
ActiveConnectionCount, N...

Count



ConsumedLCUs

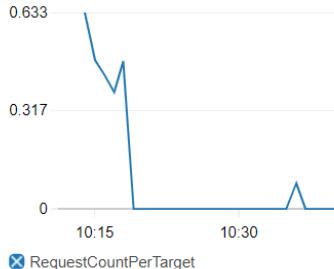
Count



Load Balancer (Qaziwa)

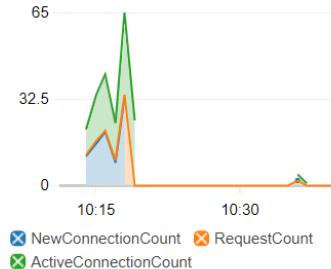
QaziwaLB_RequestCountPerT...

RequestCountPerTarget



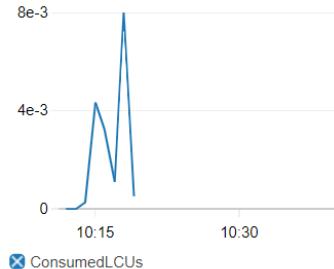
ActiveConnectionCount, NewC...

Count



ConsumedLCUs

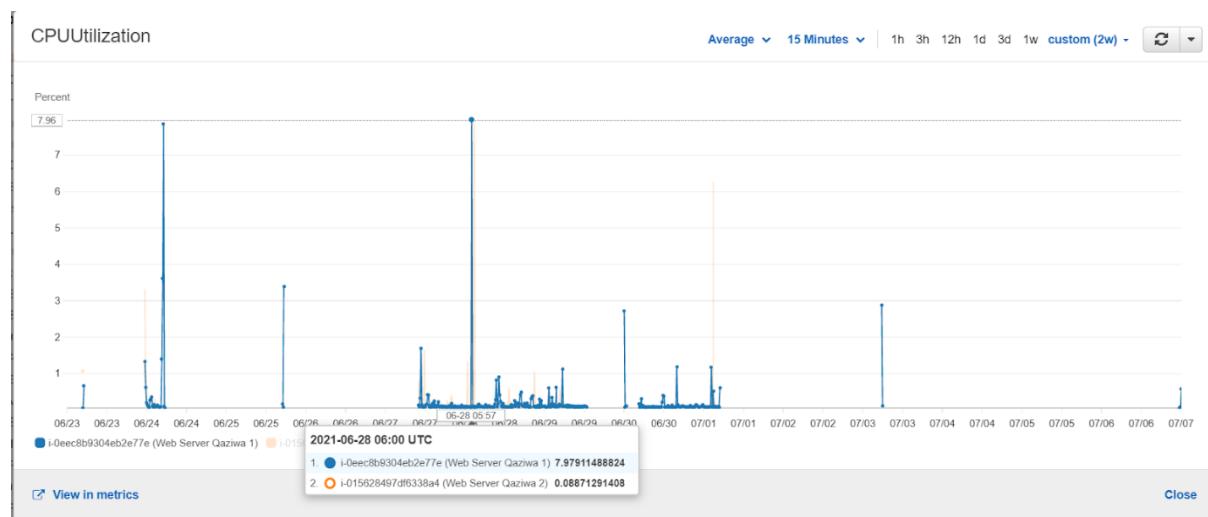
Count



In this section, we will be showing how the monitoring works across different duration.

Web Server 1 CPU Data (2 weeks)

A surge will be observed when the demand is high. From this graph, it can be observed that web server 1 has a huge surge in demand whereas the demand remains low for web server 2. This is because user traffic will first be routed to the main server which is web server 1. When the CPU utilization met the limit (reach full capacity), any new users will be directed to web server 2 as part of load balancing.



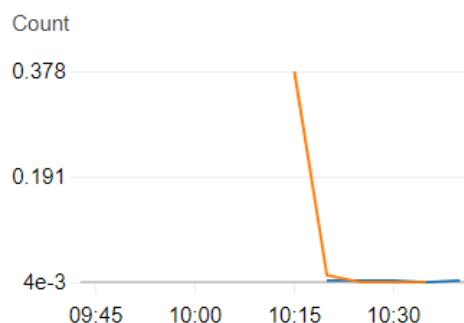
Web Server (Qaziwa)

CPUUtilization



- i-0eec8b9304eb2e77e (Web Server Qaziwa 1)
- i-015628497df6338a4 (Web Server Qaziwa 2)

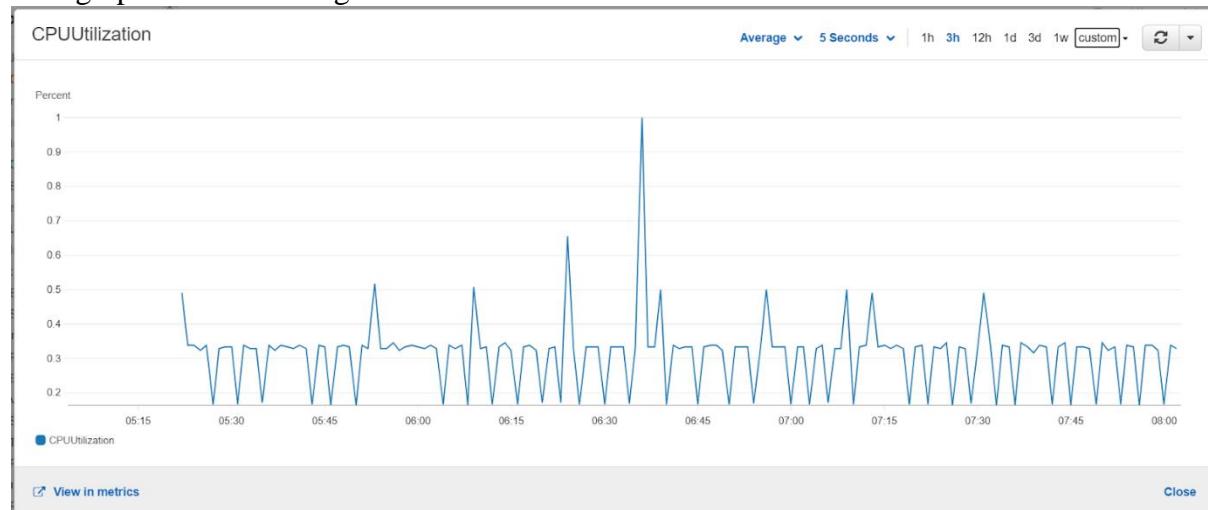
CPUCreditUsage



- i-0eec8b9304eb2e77e (Web Server Qaziwa 1)
- i-015628497df6338a4 (Web Server Qaziwa 2)

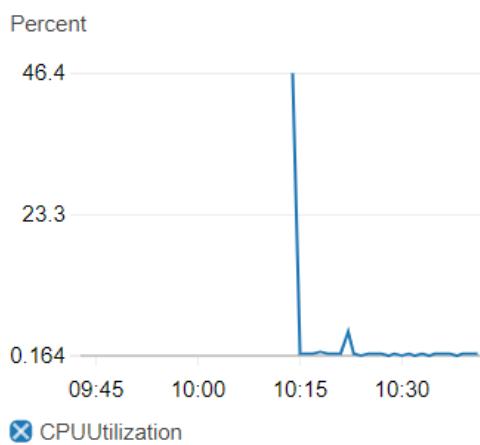
Database Server CPU Data (3 days)

This graph shows the usage rate of the database server.

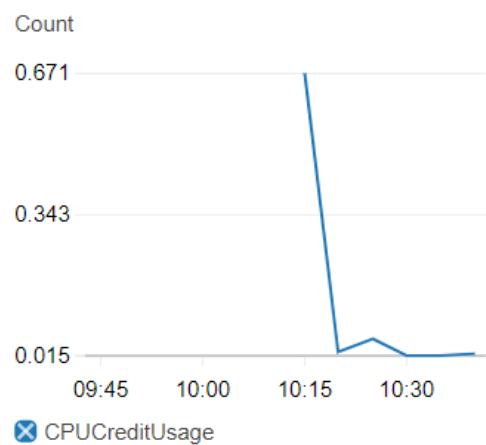


Database Server (Qaziwa)

CPUUtilization



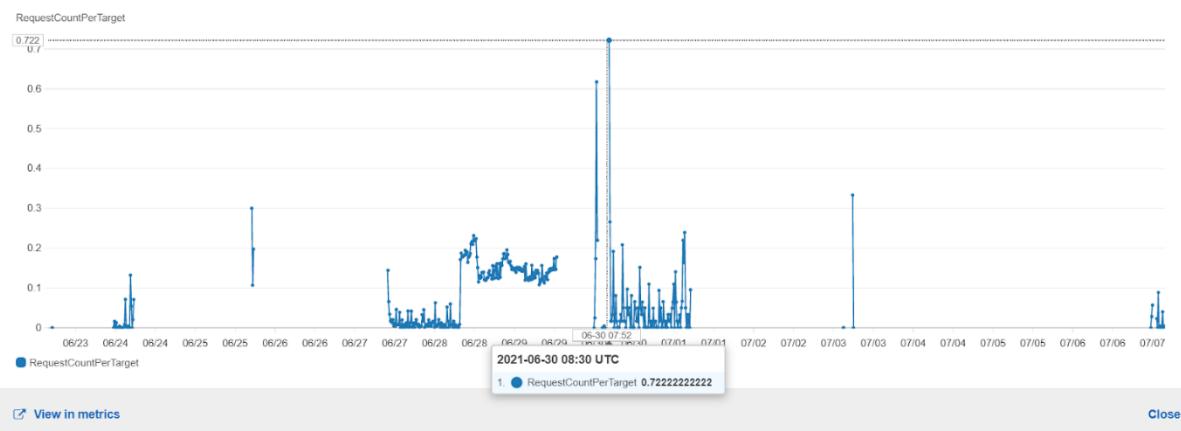
CPUCreditUsage



Application Load Balancer Data (2 weeks)

QaziwaLB_RequestCountPerTarget

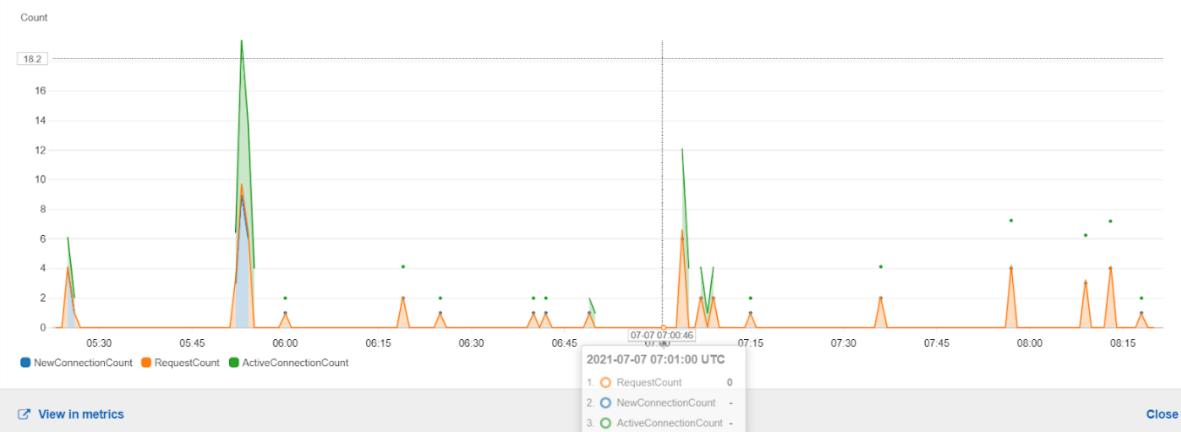
Average ▾ 15 Minutes ▾ | 1h 3h 12h 1d 3d 1w custom (2w) -



Active connection Count for Load Balancer (3 hours)

ActiveConnectionCount, NewConnectionCount, RequestCount

Average ▾ 5 Seconds ▾ | 1h 5m 12h 1d 3d 1w custom -



Notification sent via email (Wafi)

Notification

Alarm state trigger
Define the alarm state that will trigger this action.

In alarm
The metric or expression is outside of the defined threshold.

OK
The metric or expression is within the defined threshold.

Insufficient data
The alarm has just started or not enough data is available.

Remove

Select an SNS topic
Define the SNS (Simple Notification Service) topic that will receive the notification.

Select an existing SNS topic

Create new topic

Use topic ARN

Send a notification to...

vocAlarmTopic X

Only email lists for this account are available.

Email (endpoints)
19031563@myrp.edu.sg and 1 more - [View in SNS Console](#)

Add notification

Load Balancer description

Add name and description

Name and description

Alarm name
 LoadBalancerCount

Alarm description - optional
 Alarm Alerts if load balancer exceeds 1.0 count|

Up to 1024 characters (49/1024)

Cancel **Previous** **Next**

Alarms

Details of an actual alarm are shown below. After the alarm has triggered, it will notify the email address listed with a full summary of the alarm. With this alarm, it allows us to conduct troubleshooting accordingly.

You are receiving this email because your Amazon CloudWatch Alarm "Test" in the US East (N. Virginia) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [3.0 (01/07/21 07:21:00)] was greater than or equal to the threshold (2.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Thursday 01 July, 2021 07:23:15 UTC".

Alarm Details:

- Name: Test
- Description:
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [3.0 (01/07/21 07:21:00)] was greater than or equal to the threshold (2.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Thursday 01 July, 2021 07:23:15 UTC
- AWS Account: 501144450828
- Alarm Arn: arn:aws:cloudwatch:us-east-1:501144450828:alarm:Test

Threshold:

- The alarm is in the ALARM state when the metric is GreaterThanOrEqualToThreshold 2.0 for 60 seconds.

Monitored Metric:

- MetricNamespace: AWS/ApplicationELB
- MetricName: RequestCountPerTarget
- Dimensions: [TargetGroup = targetgroup/QaziwaTargetGroup/b99f92eb233fa95a]
- Period: 60 seconds
- Statistic: Sum
- Unit: not specified
- TreatMissingData: missing

State Change Actions:

- OK:
- ALARM: [arn:aws:sns:us-east-1:501144450828:vocAlarmTopic]
- INSUFFICIENT_DATA:

- Simple Notification Service (SNS)

This service is mainly based on CloudWatch. In our configuration, we will conduct a test to see whether a notification will be delivered when an abnormality is found. An endpoint needs to be specified for this to work, so we assigned Wafi's email address as a target. A notification is sent which means the alarm works successfully (example of email notification under Cloud Watch section above and Cloud Trail section below)

The screenshot shows the AWS SNS Topics page with 'vocAlarmTopic' selected. The top navigation bar includes 'Amazon SNS > Topics > vocAlarmTopic'. Below the navigation, there are three buttons: 'Edit', 'Delete', and 'Publish message'. The main content area is titled 'Details' and contains the following information:

Name	Display name
vocAlarmTopic	-
ARN	Topic owner
arn:aws:sns:us-east-1:501144450828:vocAlarmTopic	501144450828
Type	
Standard	

Subscription: d329d2c9-cccb-4f40-ad50-b01f2f0a1906

[Edit](#)[Delete](#)

Details

ARN
arn:aws:sns:us-east-1:501144450828:vocAlarmTopic:d329d2c9-cccb-4f40-ad50-b01f2f0a1906

Status
 Confirmed

Endpoint
19031563@myrp.edu.sg

Protocol
EMAIL

Topic
[vocAlarmTopic](#)

Since we already finished on our CloudTrail, now is the time to configure the CloudTrail.

Creation of CloudTrail:

While Cloud Watch focuses on mainly the activities of AWS services and resources, Cloud Trail service will record the activities made inside the AWS infrastructure and deliver log files to the S3 bucket. Event history will reflect actions including using command-line tools and different AWS services.

The cloud trail we created is called “catch-all-trails”. We also created a new S3 bucket “qaziwa-cloudtrail-logs” to store the logs. The log file validation is also enabled to ensure integrity where the contents inside the log file have not been changed. When enabled, a hash will be created for every log file delivered. In an interval for an hour, a digest file that contains references to the log file with hash is created. The digest file is associated with the cloud trail and will be delivered to other accounts when required.

CloudTrail > Dashboard > arn:aws:cloudtrail:us-east-1:501144450828:trail/catch-all-trails

catch-all-trails

[Delete](#) [Stop logging](#) [Edit](#)

General details			
Trail logging Logging	Trail log location qaziwa-cloudtrail-logs/AWSLogs/501144450828	Log file validation Enabled	SNS notification delivery Disabled
Trail name catch-all-trails	Last log file delivered July 14, 2021, 11:43:33 (UTC+08:00)	Last file validation delivered -	Last SNS notification -
Multi-region trail Yes	Log file SSE-KMS encryption Not enabled		
Apply trail to my organization Not enabled			

SNS will also be enabled where alarms will notify the listed email address (Wafi’s account) using the existing SNS topic “vocAlarmTopic”.

▼ Additional settings

Log file validation [Info](#)

Enabled

SNS notification delivery [Info](#)

Enabled

Create a new SNS topic

New

Existing

SNS topic

×

Leave management events as default

Management events [Edit](#)

API activity All	Exclude AWS KMS events No
	Exclude Amazon RDS Data API events No

Management events [Info](#)

Management events show information about management operations performed on resources in your AWS account.

i No additional charges apply to log management events on this trail because this is your first copy of management events.

API activity
Choose the activities you want to log.

Read Write

Exclude AWS KMS events

Exclude Amazon RDS Data API events

Enable insights events by enabling API call rate which will record the number of requests to the network.

Edit arn:aws:cloudtrail:us-east-1:501144450828:trail/catch-all-trails

Events Info

Record API activity for individual resources, or for all current and future resources in AWS account. [Additional charges apply](#) 

Event type

Choose the type of events that you want to log.

Insights events

Identify unusual activity, errors, or user behavior in your account.

Insights events Info

[Additional charges apply](#)  Identify unusual activity, errors, or user behavior in your account.

Choose Insights types

Insights measure unusual activity against a seven-day baseline.

API call rate

A measurement of write-only management API calls that occur per minute against a baseline API call volume.

Email notification (Wafi) - Example

Since we have a notification set to vocalarm topic, wafi's email is registered under that group and therefore would receive notifications regarding the created cloud trail if any logs are stored in the s3 bucket. An email would be sent for every log that is transferred to the S3 bucket therefore the notification traffic would be quite heavy as it is estimated that the logs are in the count of thousands.

From: [AWS Notifications](#)

Sent: Wednesday, 14 July 2021 11:08 AM

To: [MUHD WAFIYUDDIN BIN ABDUL RAHMAN](#)

Subject: AWS Notification Message

CAUTION: This email originated from outside the organization. Do not click links or open attachments unless you recognize the sender and know the content is safe.

{"s3Bucket":"aws-cloudtrail-logs-501144450828-1ec5e12a","s3ObjectKey":["AWSLogs/501144450828/CloudTrail/eu-west-1/2021/07/14/501144450828_CloudTrail_eu-west-1_20210714T0305Z_JYWK5vMLjL6mXYdx.json.gz"]}

--

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:

[https://apc01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fsns.us-east-1.amazonaws.com%2Funsubscribe.html%3FSubscriptionArn%3Darn%3Aaws%3Asns%3Aus-east-1%3A501144450828%3AvocAlarmTopic%3Ad329d2c9-ccb-4f40-ad50-b01f20a1906%26Endpoint%3D19031563%3A40myrp.edu.sg%7C3edd5e6956044ede9b3e08d94674b398%7Cf688b0d079f040a4864435fcdee9d0f3%7C0%7C637618289320109428%7CUnknown%7CTWFpbGZsb3d8eyJWljoimC4wLjAwMDAiLCJljoIV2luMzliLCJBtil6lk1haWwiLCJXCI6Mn0%3D%7C2000&data=OH%2F8dV%2FyjTu3Ayrirf9G605Qe359oQ3rf6toZV0d4%3D&reserved=0](https://apc01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fsns.us-east-1.amazonaws.com%2Funsubscribe.html%3FSubscriptionArn%3Darn%3Aaws%3Asns%3Aus-east-1%3A501144450828%3AvocAlarmTopic%3Ad329d2c9-ccb-4f40-ad50-b01f20a1906%26Endpoint%3D19031563%3A40myrp.edu.sg&data=04%7C01%7C19031563%40myrp.edu.sg%7C3edd5e6956044ede9b3e08d94674b398%7Cf688b0d079f040a4864435fcdee9d0f3%7C0%7C637618289320109428%7CUnknown%7CTWFpbGZsb3d8eyJWljoimC4wLjAwMDAiLCJljoIV2luMzliLCJBtil6lk1haWwiLCJXCI6Mn0%3D%7C2000&data=OH%2F8dV%2FyjTu3Ayrirf9G605Qe359oQ3rf6toZV0d4%3D&reserved=0)

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at <https://apc01.safelinks.protection.outlook.com/?url=https%3A%2F%2Faws.amazon.com%2Fsupport&data=04%7C01%7C19031563%40myrp.edu.sg%7C3edd5e6956044ede9b3e08d94674b398%7Cf688b0d079f040a4864435fcdee9d0f3%7C0%7C637618289320109428%7CUnknown%7CTWFpbGZsb3d8eyJWljoimC4wLjAwMDAiLCJljoIV2luMzliLCJBtil6lk1haWwiLCJXCI6Mn0%3D%7C2000&data=Cbbi%2BBqSG3MX7dA92jWN7U7d6KA%2Fv%2F2VH%2BcVuJtrgcp%3D&reserved=0>

The event history shows all the activities achieved within this AWS account. It can be filtered accordingly from different actions done (e.g. create, modify) or across different time frames. The record of events will last for 90 days.

CloudTrail > Event history

Event history (50+) [Info](#)

Event history shows you the last 90 days of management events.

Read-only	Event name	Event time	User name	Event source	Resource type	Resource name
<input type="checkbox"/>	PutBucketPublicAcce...	July 14, 2021, 11:39:26 (UTC+0...)	user1406304=190...	s3.amazonaws.com	AWS::S3::Bucket	qaziwa-cloudtrail-logs
<input type="checkbox"/>	CreateBucket	July 14, 2021, 11:39:25 (UTC+0...)	user1406304=190...	s3.amazonaws.com	AWS::S3::Bucket	qaziwa-cloudtrail-logs
<input type="checkbox"/>	PutBucketPolicy	July 14, 2021, 11:39:25 (UTC+0...)	user1406304=190...	s3.amazonaws.com	AWS::S3::Bucket	qaziwa-cloudtrail-logs
<input type="checkbox"/>	CreateEnvironment	July 14, 2021, 11:31:14 (UTC+0...)	user1406304=190...	cloudshell.amazonaws.com	-	-
<input type="checkbox"/>	UpdateRole	July 14, 2021, 11:30:31 (UTC+0...)	vocstartsoft16262...	iam.amazonaws.com	-	-
<input type="checkbox"/>	UpdateRole	July 14, 2021, 11:30:30 (UTC+0...)	vocstartsoft16262...	iam.amazonaws.com	-	-
<input type="checkbox"/>	UpdateTrail	July 14, 2021, 11:09:12 (UTC+0...)	user1406304=190...	cloudtrail.amazonaws.com	AWS::CloudTrail::Trail, ...	arn:aws:cloudtrail:us-east-1:50114...
<input type="checkbox"/>	PutBucketPolicy	July 14, 2021, 11:09:12 (UTC+0...)	user1406304=190...	s3.amazonaws.com	AWS::S3::Bucket	aws-cloudtrail-logs-50114445082...
<input type="checkbox"/>	UpdateTrail	July 14, 2021, 11:07:00 (UTC+0...)	user1406304=190...	cloudtrail.amazonaws.com	AWS::CloudTrail::Trail, ...	arn:aws:cloudtrail:us-east-1:50114...
<input type="checkbox"/>	PutBucketPolicy	July 14, 2021, 11:07:00 (UTC+0...)	user1406304=190...	s3.amazonaws.com	AWS::S3::Bucket	aws-cloudtrail-logs-50114445082...

From the newly created S3 bucket, we can see that the logs are stored in the following screenshot.

Amazon S3 > qaziwa-cloudtrail-logs > AWSLogs/ > 501144450828/ > CloudTrail/ > us-east-1/ > 2021/ > 07/ > 14/

14/ [Copy S3 URI](#)

[Objects](#) [Properties](#)

Objects (8)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	501144450828_CloudTrail_us-east-1_20210714T0330Z_SwUk8nmndCMIPUUjr.json.gz	gz	July 14, 2021, 11:39:53 (UTC+08:00)	740.0 B	Standard
<input type="checkbox"/>	501144450828_CloudTrail_us-east-1_20210714T0330Z_UOL37Bnf5Zh5Loac.json.gz	gz	July 14, 2021, 11:39:54 (UTC+08:00)	1.5 KB	Standard
<input type="checkbox"/>	501144450828_CloudTrail_us-east-1_20210714T0345Z_Jzq2rnlllocQ03ul.json.gz	gz	July 14, 2021, 11:43:34 (UTC+08:00)	12.7 KB	Standard

3 objects were created. All logs are stored in “CloudTrail/”.

Amazon S3 > qaziwa-cloudtrail-logs > AWSLogs/ > 501144450828/

501144450828/

Objects **Properties**

Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Actions **Create folder** **Upload**

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	CloudTrail-Digest/	Folder	-	-	-
<input type="checkbox"/>	CloudTrail-Insight/	Folder	-	-	-
<input type="checkbox"/>	CloudTrail/	Folder	-	-	-

Amazon S3 > qaziwa-cloudtrail-logs > AWSLogs/ > 501144450828/ > **CloudTrail-Digest/** > us-east-1/ > 2021/ > 07/ > 14/

14/

Objects **Properties**

Objects (4)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Actions **Create folder** **Upload**

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	501144450828_CloudTrail-Digest_us-east-1Catch-all-trails_us-east-1_20210714T030700Z.json.gz	gz	July 14, 2021, 12:00:49 (UTC+08:00)	344.0 B	Standard
<input type="checkbox"/>	501144450828_CloudTrail-Digest_us-east-1Catch-all-trails_us-east-1_20210714T040700Z.json.gz	gz	July 14, 2021, 13:01:11 (UTC+08:00)	3.4 KB	Standard
<input type="checkbox"/>	501144450828_CloudTrail-Digest_us-east-1Catch-all-trails_us-east-1_20210714T050700Z.json.gz	gz	July 14, 2021, 14:01:01 (UTC+08:00)	1.1 KB	Standard

Amazon S3 > qaziwa-cloudtrail-logs > AWSLogs/ > 501144450828/ > **CloudTrail-Insight/**

CloudTrail-Insight/

Objects **Properties**

Objects (0)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Actions **Create folder** **Upload**

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
No objects					
You don't have any objects in this folder.					
Upload					

Default bucket. All the logs would immediately be sent to the default bucket if a new bucket is not created.

Amazon S3 > **aws-cloudtrail-logs-501144450828-1ec5e12a** > AWSLogs/ > 501144450828/ > CloudTrail/ > us-east-1/ > 2021/ > 07/ > 14/

14/ Copy S3 URI

Objects Properties

Objects (23)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	501144450828_CloudTrail_us-east-1_20210714T0250Z_NbN7L15loD7inTz8.json.gz	gz	July 14, 2021, 11:00:57 (UTC+08:00)	3.7 KB	Standard
<input type="checkbox"/>	501144450828_CloudTrail_us-east-1_20210714T0250Z_VXasJEngIDayPnDt.json.gz	gz	July 14, 2021, 11:03:24 (UTC+08:00)	630.0 B	Standard
<input type="checkbox"/>	501144450828_CloudTrail_us-east-1_20210714T0300Z_767ReAbUyXHGP.json.gz	gz	July 14, 2021, 11:02:26 (UTC+08:00)	926.0 B	Standard

Amazon S3 > **aws-cloudtrail-logs-501144450828-1ec5e12a** > AWSLogs/ > 501144450828/ > CloudTrail-Digest/

CloudTrail-Digest/ Copy S3 URI

Objects Properties

Objects (0)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
--------------------------	------	------	---------------	------	---------------

No objects
You don't have any objects in this folder.

Upload

Amazon S3 > **aws-cloudtrail-logs-501144450828-1ec5e12a** > AWSLogs/ > 501144450828/ > CloudTrail-Insight/

CloudTrail-Insight/ Copy S3 URI

Objects Properties

Objects (0)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
--------------------------	------	------	---------------	------	---------------

No objects
You don't have any objects in this folder.

Upload

Reference Summary of buckets

Buckets (3) Info				
Copy ARN Empty Delete Create bucket				
<input type="text"/> Find buckets by name < 1 > 				
Name	AWS Region	Access	Creation date	
aws-cloudtrail-logs-501144450828-1ec5e12a	US East (N. Virginia) us-east-1	Bucket and objects not public	July 14, 2021, 10:56:52 (UTC+08:00)	
qaziwa-cloudtrail-logs	US East (N. Virginia) us-east-1	Bucket and objects not public	July 14, 2021, 11:39:26 (UTC+08:00)	
qaziwabucket	US East (N. Virginia) us-east-1	Objects can be public	June 2, 2021, 13:40:38 (UTC+08:00)	

Since we have finished configuring the CloudTrail, now we can continue to create the IAM roles and groups.

Creation of IAMs:

With all the instances and security configurations put into place, we would now need to create restrictions for the users that would be accessing the AWS cloud. Taking a look at the bottom screenshot the end result would be 3 created users configured for access to the cloud each with varying permissions set.

All Users:

IAM users (3) Info						
Delete Add users						
<input type="text"/> Search < 1 > 						
User name	Groups	Last activity	MFA	Console last sign...	Access key age	
qaziwa-admin	0	Never	None	None	-	
Qaziwa-Developer	0	Never	None	None	-	
Qaziwa-NormalUser	0	Never	None	None	-	

The first user that we would configure for use would be the Admin user. This user would have the most privileges out of the three as this is essentially the placeholder for the root account. Because of this, in the permission policy, we would set admin to have AdministratorAccess. This allows the account to have access to all the AWS tools like the root account has.

Qaziwa-Admin

Creation time: 2021-06-18 14:55 UTC+0800

Permissions | Groups | Tags (1) | Security credentials | Access Advisor

Permissions policies (1 policy applied)

Add permissions | Add inline policy

Policy name	Policy type
AdministratorAccess	AWS managed policy

Policy summary | { } JSON | Simulate policy

Filter

Service	Access level	Resource	Request condition
Access Analyzer	Full access	All resources	None
Account	Full access	All resources	None
Activate	Full access	All resources	None
Alexa for Business	Full access	All resources	None

This user focuses on creating and managing the instances that are on the cloud. This user is the developer and should be allowed to have read and write on services relating to the instance. We would set it for AmazonEC2FullAccess, CloudwatchFullAccess, CloudTrailReadOnly.

EC2FullAccess: Any function regarding the instance under EC2 is given read and write access.

CloudwatchFullAccess: User is able to read the monitoring dashboard and add their own variables to monitor or edit the value type it returns.

Qaziwa-Developer

Users > Qaziwa-Developer

Summary | Delete user | ?

User ARN: arn:aws:iam::501144450828:user/Qaziwa-Developer | Path: / | Creation time: 2021-06-18 15:05 UTC+0800

Permissions | Groups | Tags (1) | Security credentials | Access Advisor

Permissions policies (3 policies applied)

Add permissions | Add inline policy

Policy name	Policy type
AmazonEC2FullAccess	AWS managed policy
CloudWatchFullAccess	AWS managed policy
AWSCloudTrailReadOnlyAccess	AWS managed policy

Permissions boundary (not set)

Qaziwa-NormalUser

Normal user would be the user that uses the cloud to host their services like the clients for example. Because of this they are given the lowest privilege only allowing the ability to connect and view the instance and list the instance.

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*

Qaziwa-NormalUser

[+ Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type*

Programmatic access

Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

AWS Management Console access

Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password*

Autogenerated password

Custom password

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type*

Programmatic access

Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

AWS Management Console access

Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password*

Autogenerated password

Custom password

Require password reset

User must create a new password at next sign-in

Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

* Required

[Cancel](#)

[Next: Permissions](#)

Add user to group existing user directly

Create policy Refresh

Filter policies ▾ Q ec2 ▾ Showing 25 results

	Policy name	Type	Used as
...	EBSImageBuilderCrossAccountDistributionPolicy	AWS managed	None
<input checked="" type="checkbox"/> EC2InstanceConnect	EC2InstanceConnect	AWS managed	None

EC2InstanceConnect
Allows customers to call EC2 Instance Connect to publish ephemeral keys to their EC2 instances and connect via ssh or the EC2 Instance Connect CLI.

Policy summary { } JSON

Filter

Service	Access level	Resource	Request condition
---------	--------------	----------	-------------------

▶ Set permissions boundary

Add user to group existing user directly

Create policy Cancel Previous Next: Tags

Filter policies ▾ Q ec2 ▾ Showing 25 results

	Policy name	Type	Used as
...	EBSImageBuilderCrossAccountDistributionPolicy	AWS managed	None
<input checked="" type="checkbox"/> EC2InstanceConnect	EC2InstanceConnect	AWS managed	None

Policy summary { } JSON

Filter

Service	Access level	Resource	Request condition
---------	--------------	----------	-------------------

Allow (2 of 285 services) Show remaining 283

EC2	Limited: List	All resources	None
EC2 Instance Connect	Limited: Write	All resources	None

▶ Set permissions boundary

Cancel Previous Next: Tags

Add user to group | existing user | directly

Create policy

Filter policies ▾ ec2 Showing 25 results

	Policy name ▾	Type	Used as
<input type="checkbox"/>	AWSApplicationMigrationEC2Access	AWS managed	None
<input type="checkbox"/>	AWSElasticBeanstalkCustomPlatformforEC2Role	AWS managed	None
<input type="checkbox"/>	AWSOpsWorksRegisterCLI_EC2	AWS managed	None
<input type="checkbox"/>	CloudWatchActionsEC2Access	AWS managed	None
<input type="checkbox"/>	Ec2ImageBuilderCrossAccountDistributionAccess	AWS managed	None
<input checked="" type="checkbox"/>	EC2InstanceConnect	AWS managed	None
<input type="checkbox"/>	EC2InstanceProfileForImageBuilder	AWS managed	None
<input type="checkbox"/>	EC2InstanceProfileForImageBuilderECRContainerBuilds	AWS managed	None

Set permissions boundary

Cancel Previous Next: Tags

Add user

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	Qaziwa-NormalUser
AWS access type	AWS Management Console access - with a password
Console password type	Autogenerated
Require password reset	Yes
Permissions boundary	Permissions boundary is not set

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	EC2InstanceConnect
Managed policy	IAMUserChangePassword

Tags

The new user will receive the following tag

Key	Value
Name:	NormalUser

Cancel Previous Create user

AWS educate restriction limits allocation of users.

You successfully created the following users, but some problems remain. Expand the following section for details. View and download user's security credentials as well as email users instructions to log into the AWS Management Console. This will be the last time these credentials will be available to download. However, you can manage and recreate these credentials at any time.

Users with AWS Management Console access may sign-in at: <https://501144450828.signin.aws.amazon.com/console>

 Download .csv

User

▼  Qaziwa-NormalUser

- ✓ Created user Qaziwa-NormalUser
- ✓ Attached policy EC2InstanceConnect to user Qaziwa-NormalUser
- ✓ Attached policy IAMUserChangePassword to user Qaziwa-NormalUser

❗ Could not create login profile for user Qaziwa-NormalUser:

User: arn:aws:sts::501144450828:assumed-role/vocstartsoft/user1406304=19031563@myrp.edu.sg is not authorized to perform: iam>CreateLoginProfile

Close