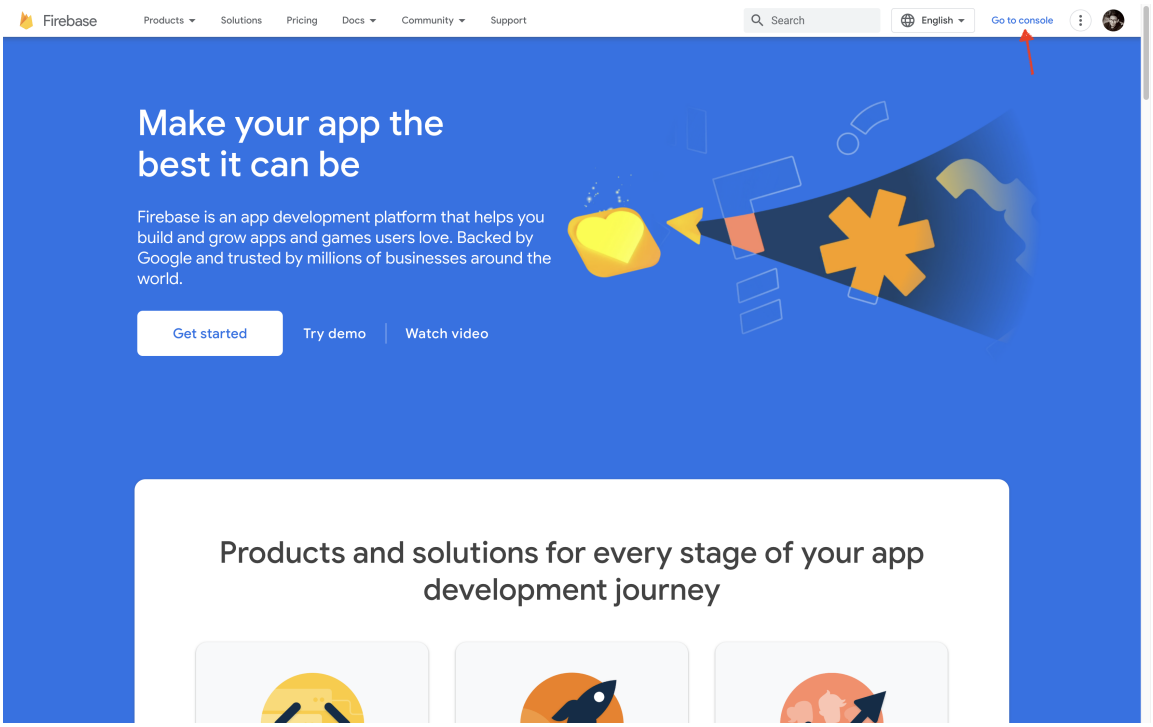
Steps to follow.

Hello, Thanks for buying the source code. Since you got your source code for “blogging website”. You need to do some things before you can access it on your browser. If you don’t follow the steps mentioned below then you won’t be able to run the site.

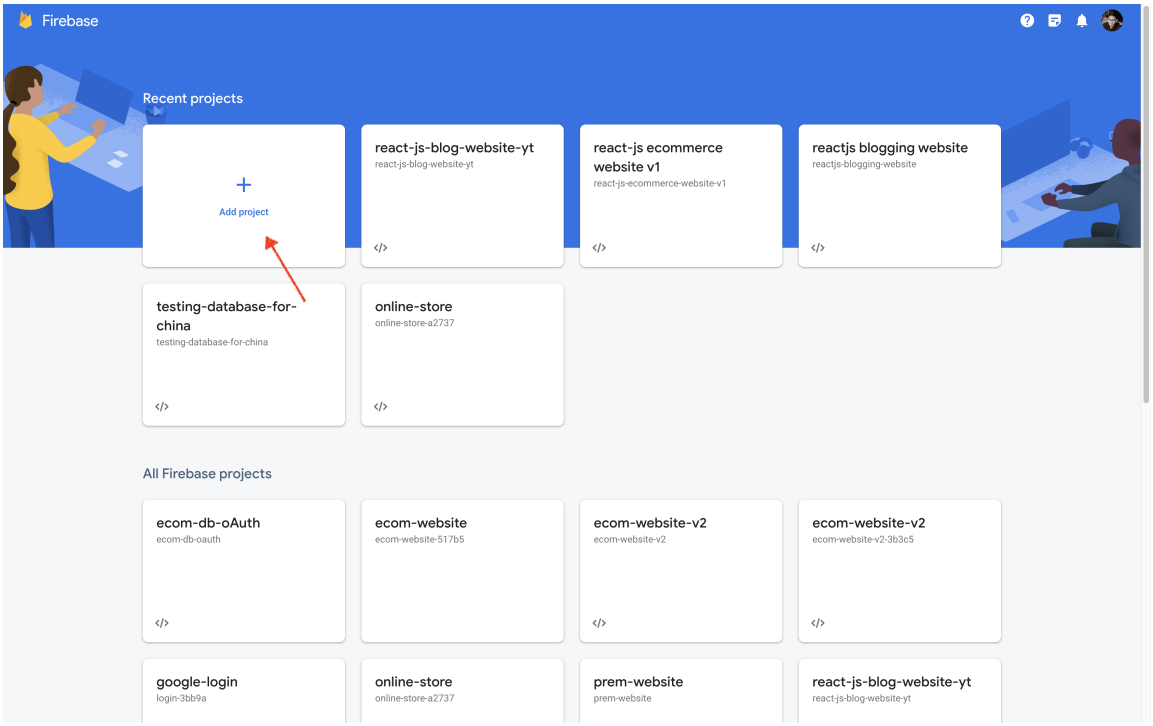
1. Open “blogging website” folder either in you VSCode terminal or any terminal of your choice.
2. In terminal once you are inside “blogging website” folder. Run `npm install` command

   Note - If it gives error make sure you have node.js and npm installed in your system. You can check that by running `npm -v` and `node -v` cmd both cmd will give you version numbers. If you get error install nodejs from below and install it in you system. Then restart your terminal inside the folder.

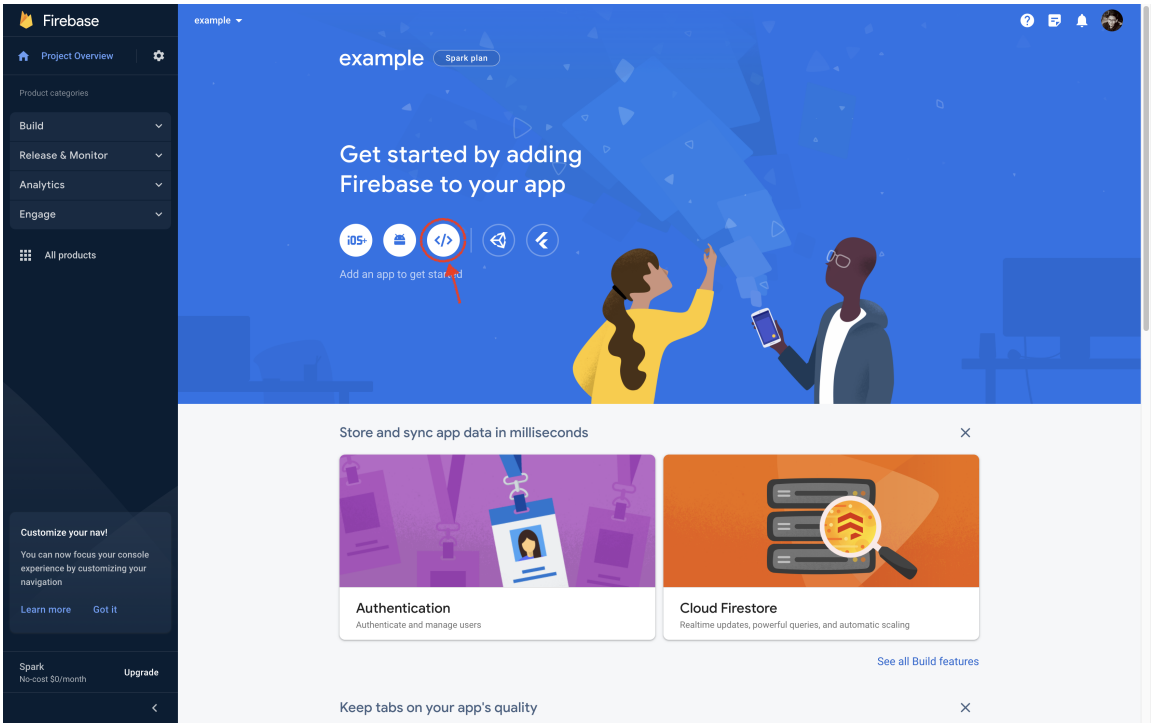
   Download NodeJS - https://nodejs.org/en
3. Once all the packages install. We need to setup our firebase. So go to https://firebase.google.com/
4. Log in or Create an account
5. Then if you are not in your console click on “go to console” button



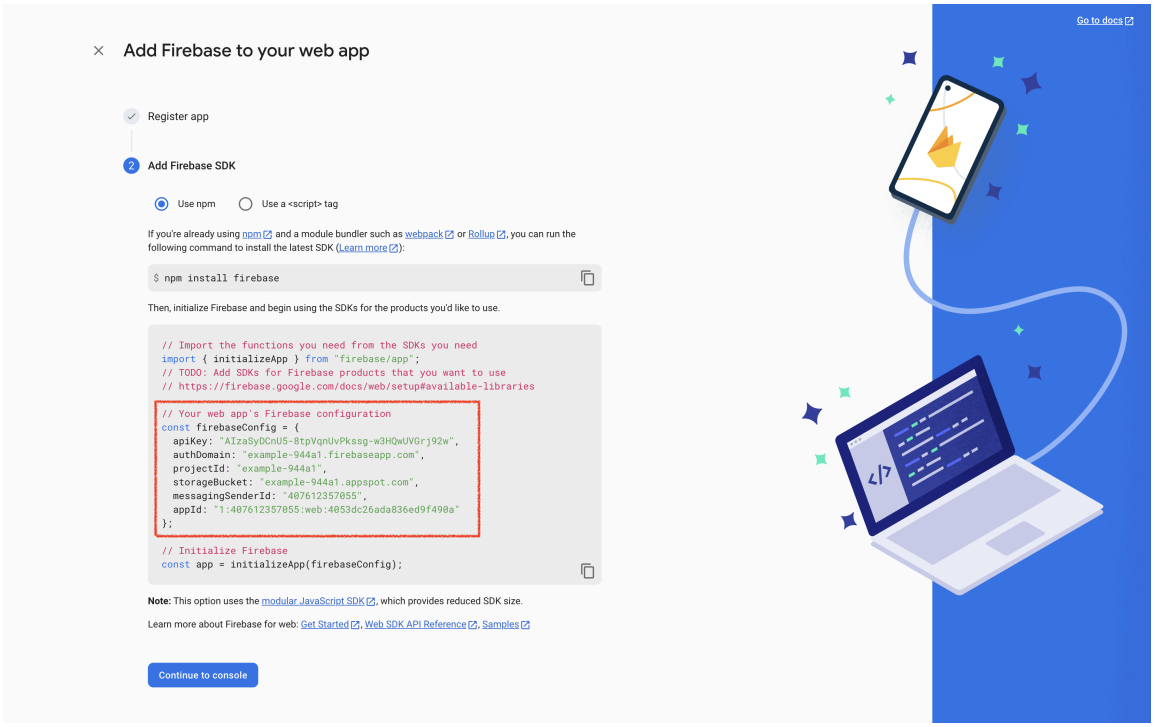
6. Click on “new project”



7. Enter “Project Name” & then click “continue”
8. Turn off “google analytics” and then “create project”. It will take some time to create the project.
9. Click on “web” icon & register you app with any name.



10. Copy whole “firebaseConfig” object.



11. Then open “firebase.jsx” file inside “blogging website/src/common” folder & replace the “firebaseConfig” variable with the one that you just copied from your project.

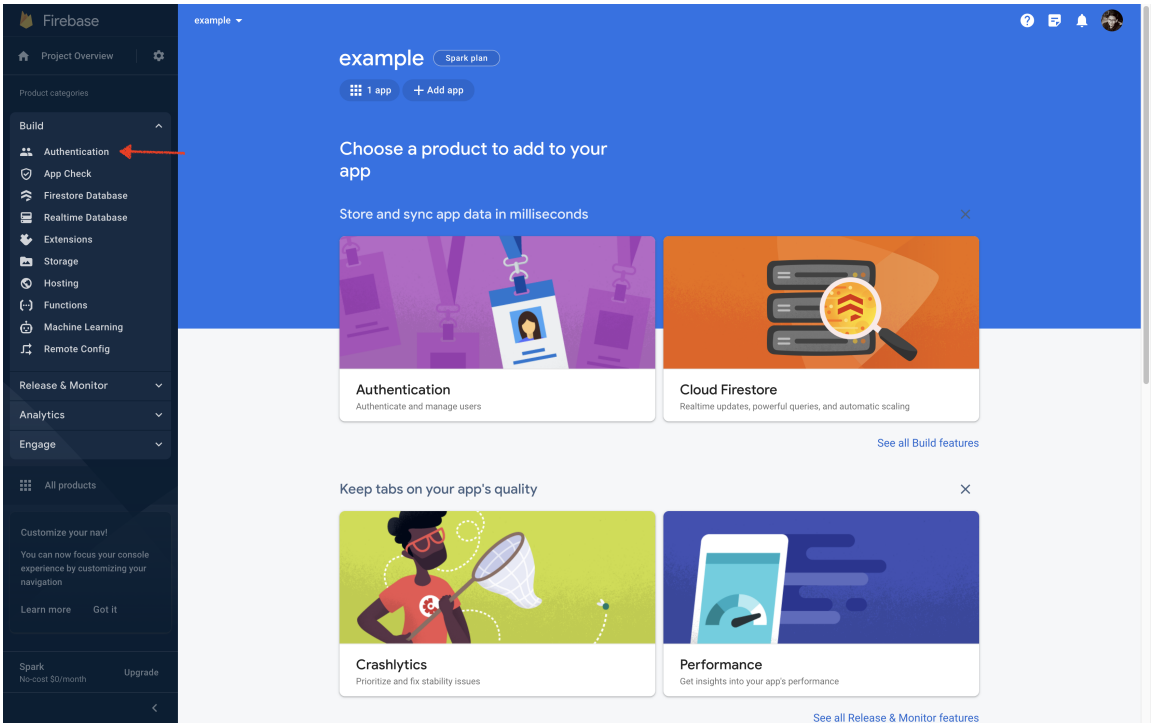
    Before

```
const firebaseConfig = {
    // replace this object with you project firebaseConfig.
};
```

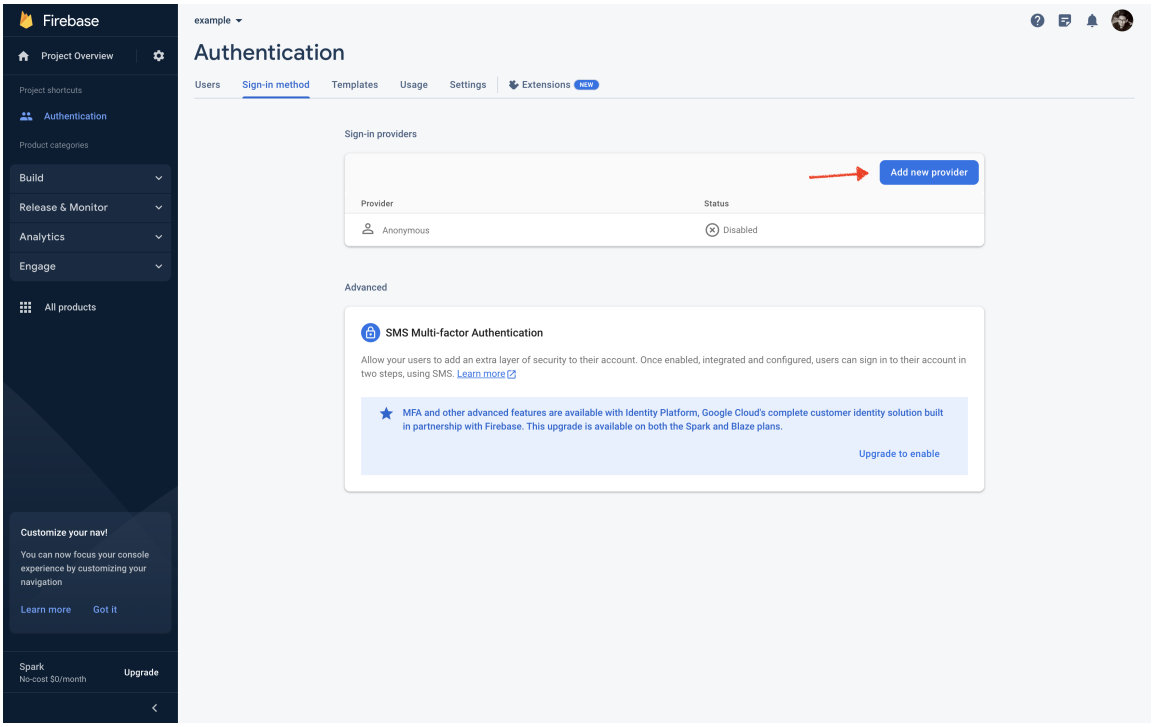
After

```
const firebaseConfig = {
    // below values will be different for your firebase project
    apiKey: "AIzaSyDCnU5-8tpVqnUvPkssg-w3HQwUVGrj92w",
    authDomain: "example-944a1.firebaseapp.com",
    projectId: "example-944a1",
    storageBucket: "example-944a1.appspot.com",
    messagingSenderId: "407612357055",
    appId: "1:407612357055:web:4053dc26ada836ed9f490a"
};
```

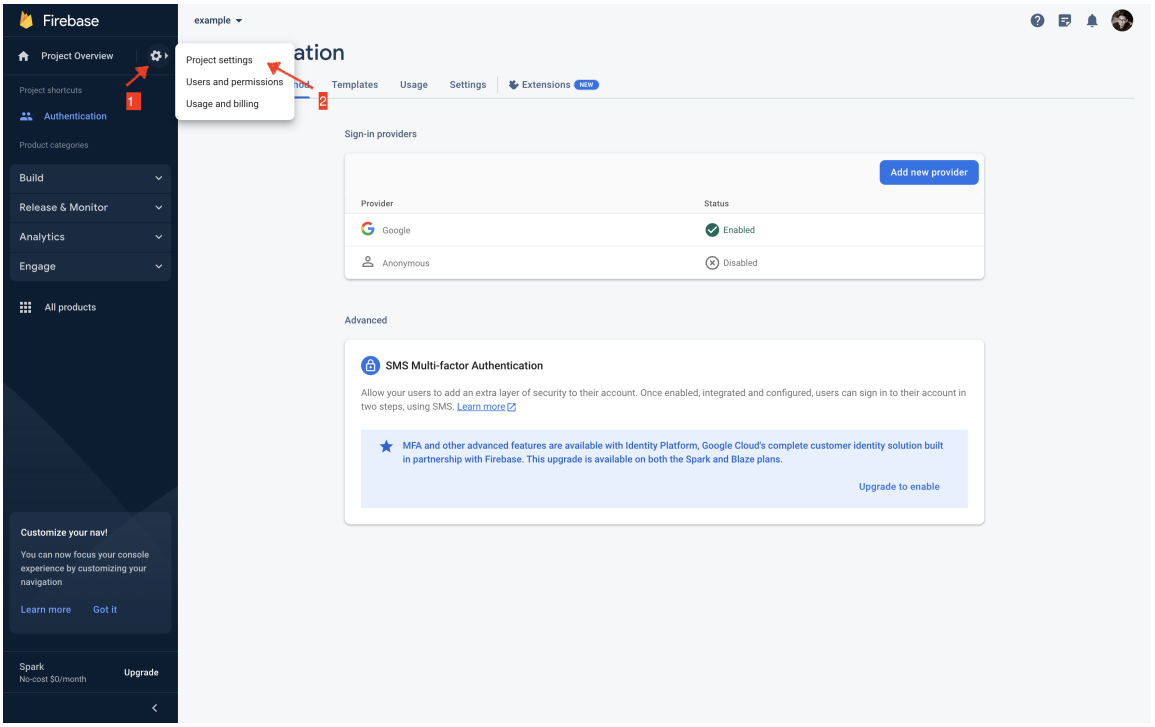
12. Now open your “server” folder in a new terminal & run `npm install` cmd.
13. Now go back to you firebase & go to “Authentication” now.



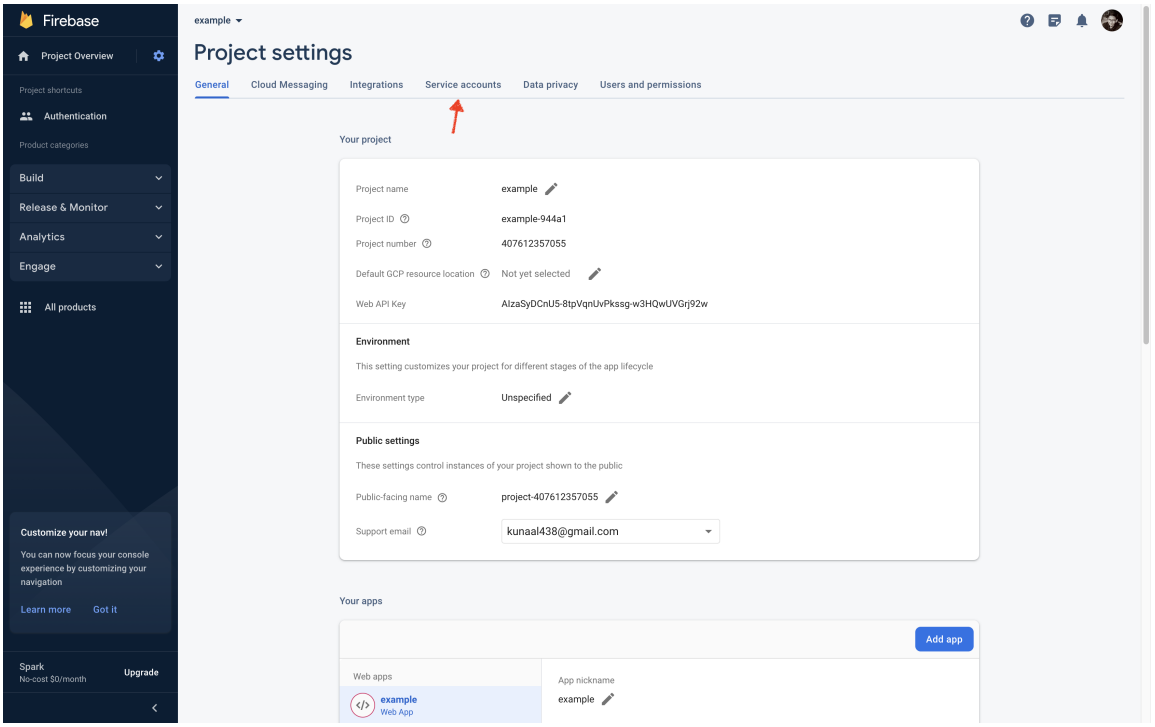
14. Click on “get Started”
15. Now click on “Add new provider”



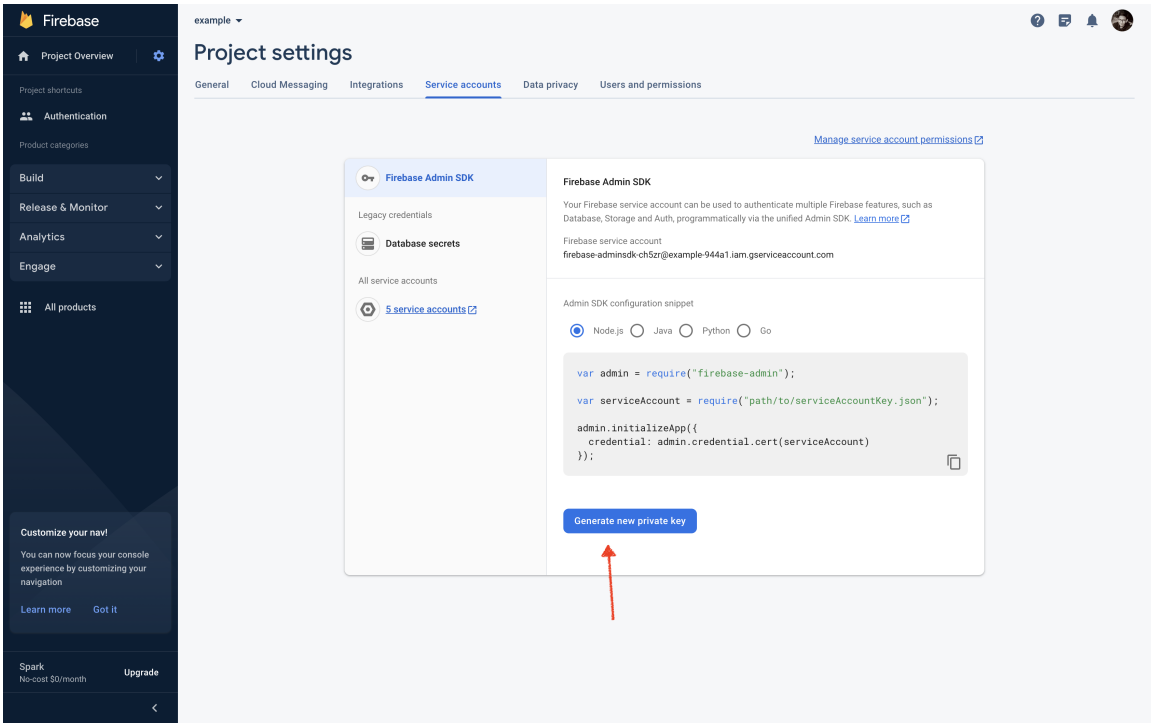
16. Select “google” from the list of options. Turn on “Enable” and select your “email” in “Support email for project”
17. Click on save. It will add “google authentication” to your project.
18. Now, go to “project’s settings”



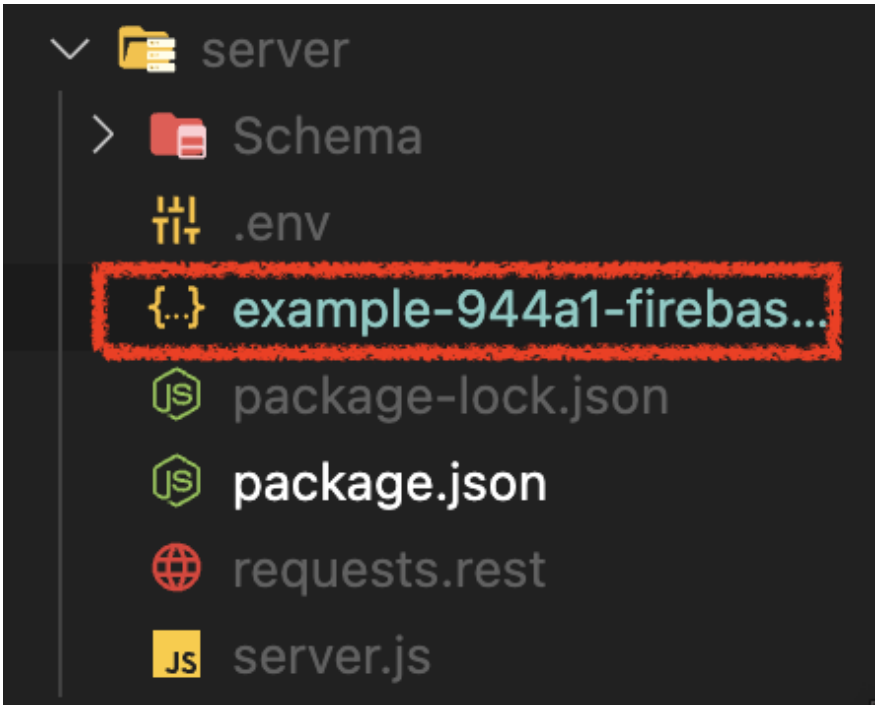
19. Go to “service accounts”.



20. Then “generate new private key”. It will download a JSON file in your system.



21. Cut the downloaded JSON file and paste it inside “server” folder.



22. Open “server.js” now and add your JSON file on line no. 33 inside “ “

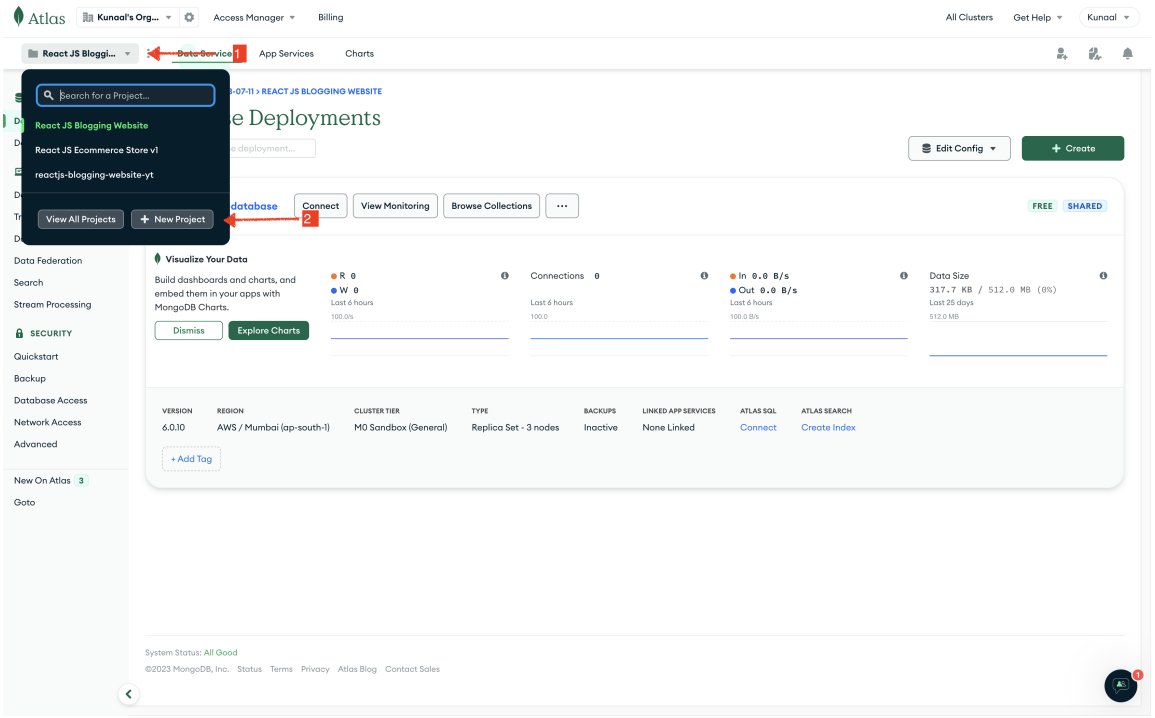
    Before

```
33  import serviceAccount from "your firebase generated file.json" assert { type: "json" };
```

    After

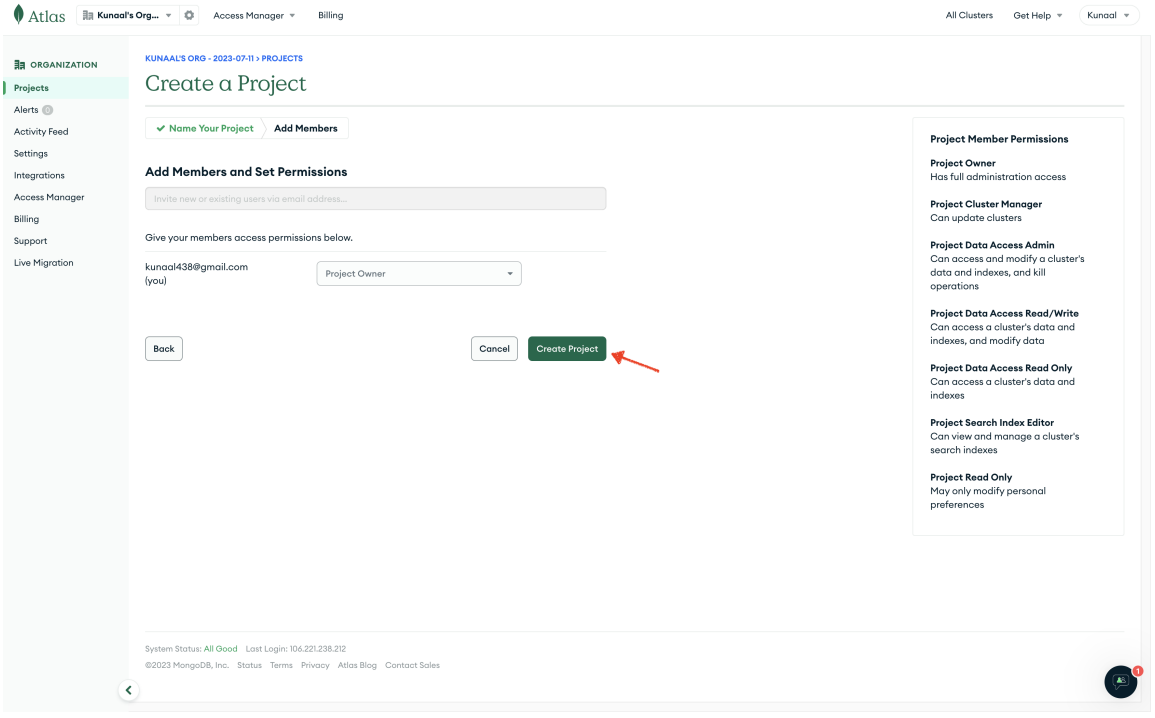
```
33  import serviceAccount from "./example-944a1-firebase-adminsdk-ch5zr-3cb9422ea8.json" assert { type: "json" };
```

23. Your Firebase is connected. Now let’s connect MongoDB
24. Go to - https://www.mongodb.com/
25. Sign in or create an account. If you create account then jump directly to step 30.
26. Now create a “new project”

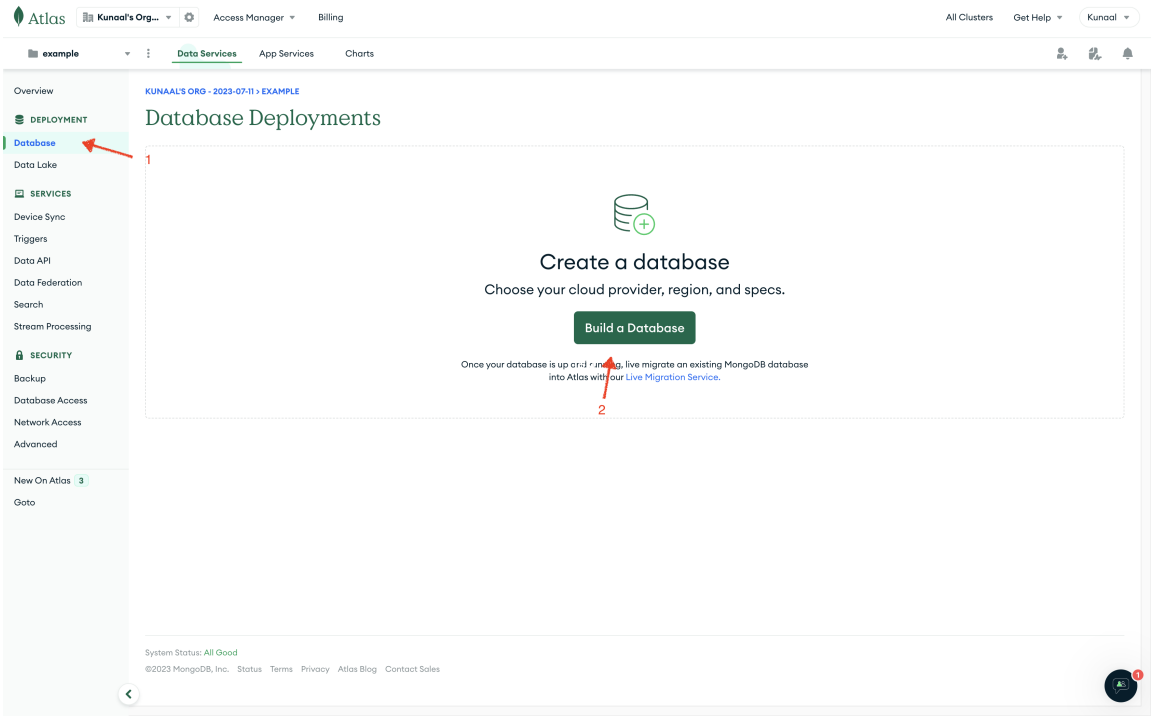


27. Give a project name.

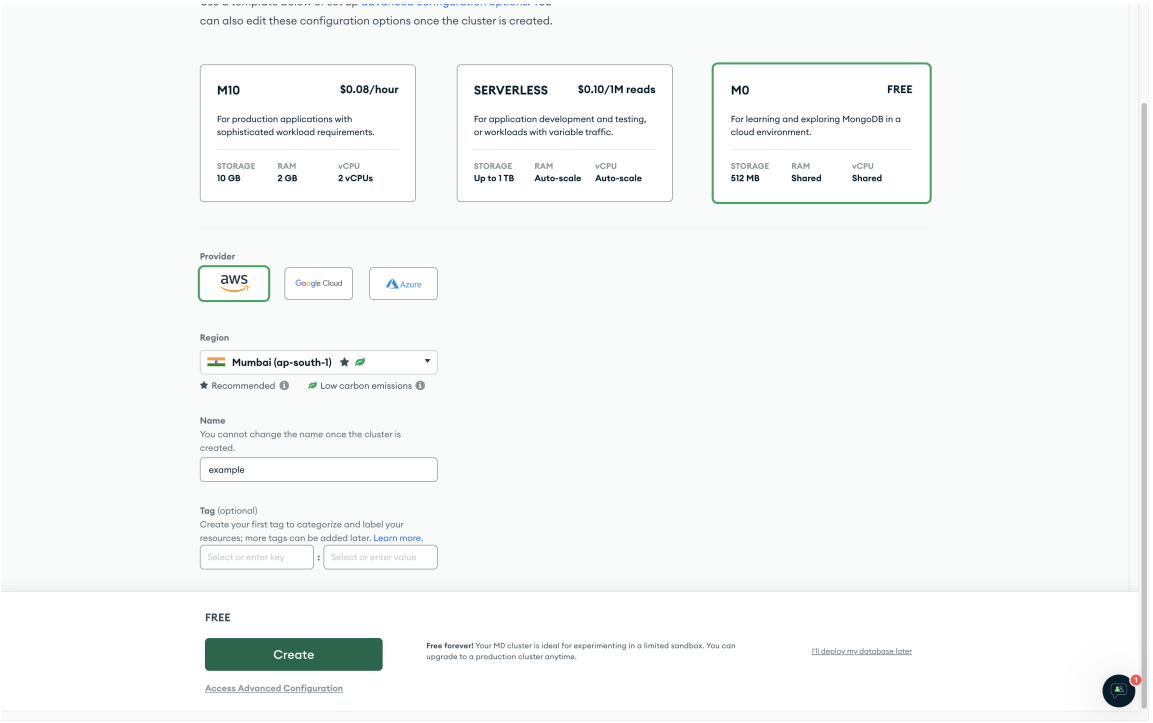
28. Since you don't have any member. Click on "create project"



29. Now go to database & create a new database.



30. Select FREE tier.
31. Choose any server provider. Mine is AWS
32. In region Choose whatever region is nearest to you.
33. Give a name to cluster & Click “create”

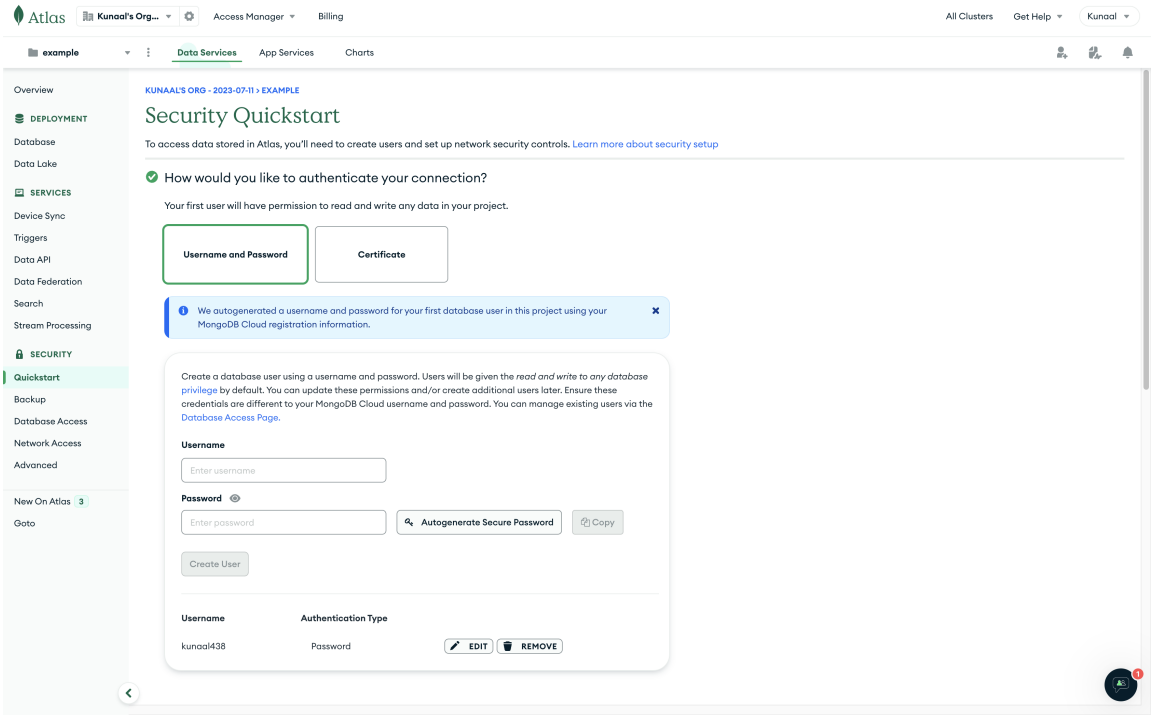


34. You will get redirected to “Security QuickStart” page.

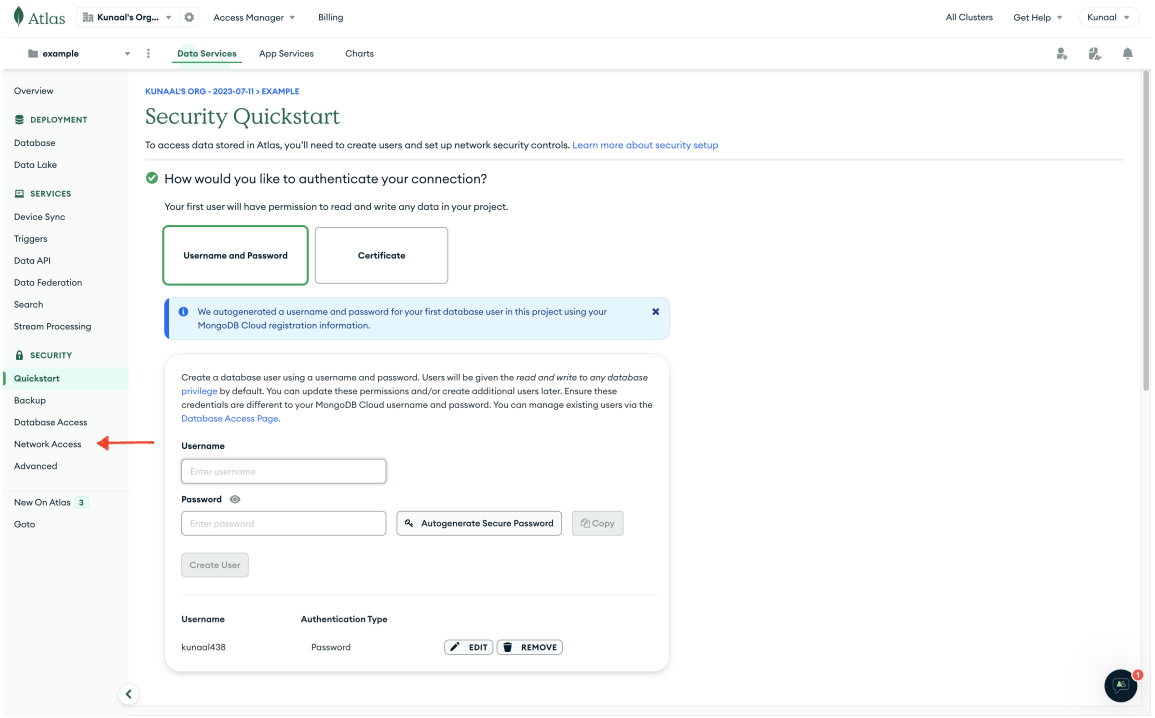
35. First select the “username and password” option. Under that give a username of your choice and password. If you want to autogenerate password. Click on autogenerate password on the right side of “password” field.

    Note - Copy your user’s password and paste it somewhere because if its a autogenerated password you can see it only once. So copy it and paste it somewhere before creating a user.

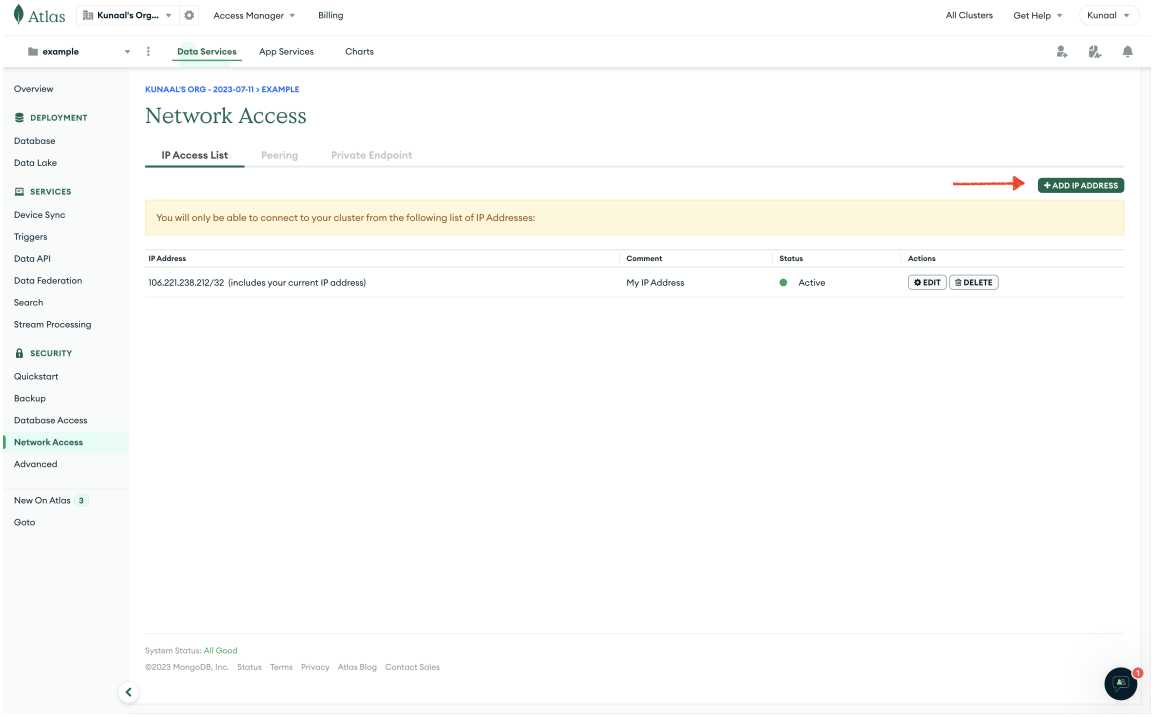
    Note - If you didn’t copy the password before creating a user. Then in the list of user’s click on “edit” icon and generate new password again and this time copy the password somewhere first before updating the password.



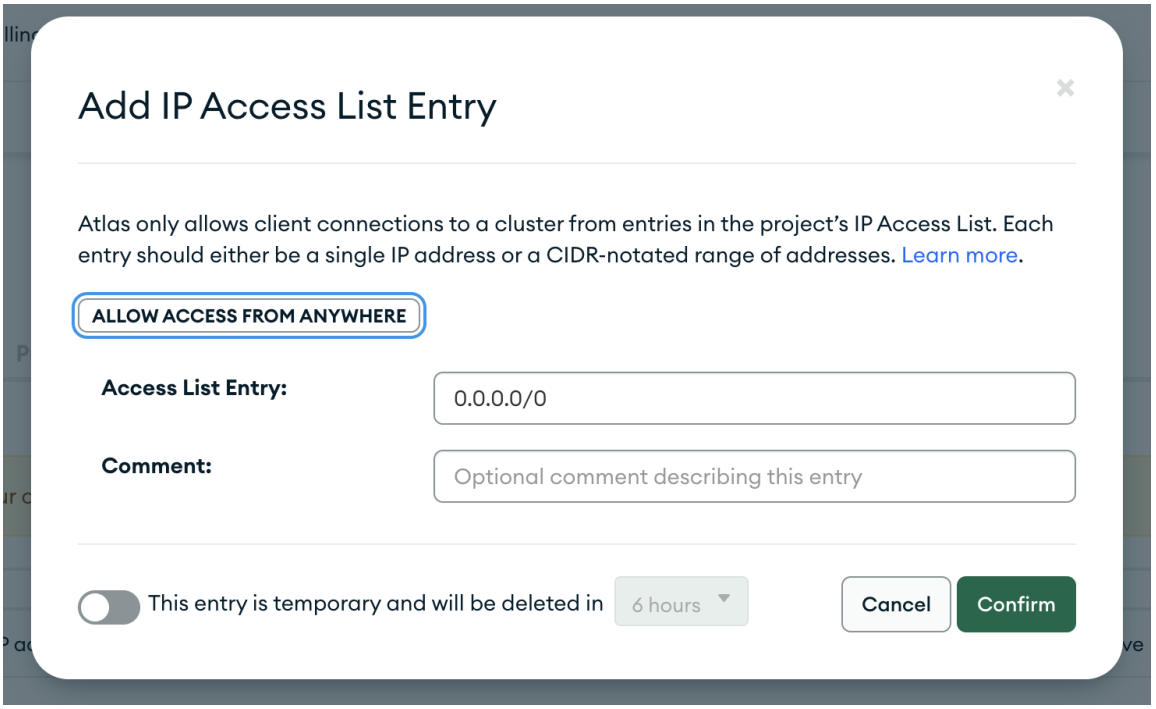
36. Now go to “Network Access” under “security”



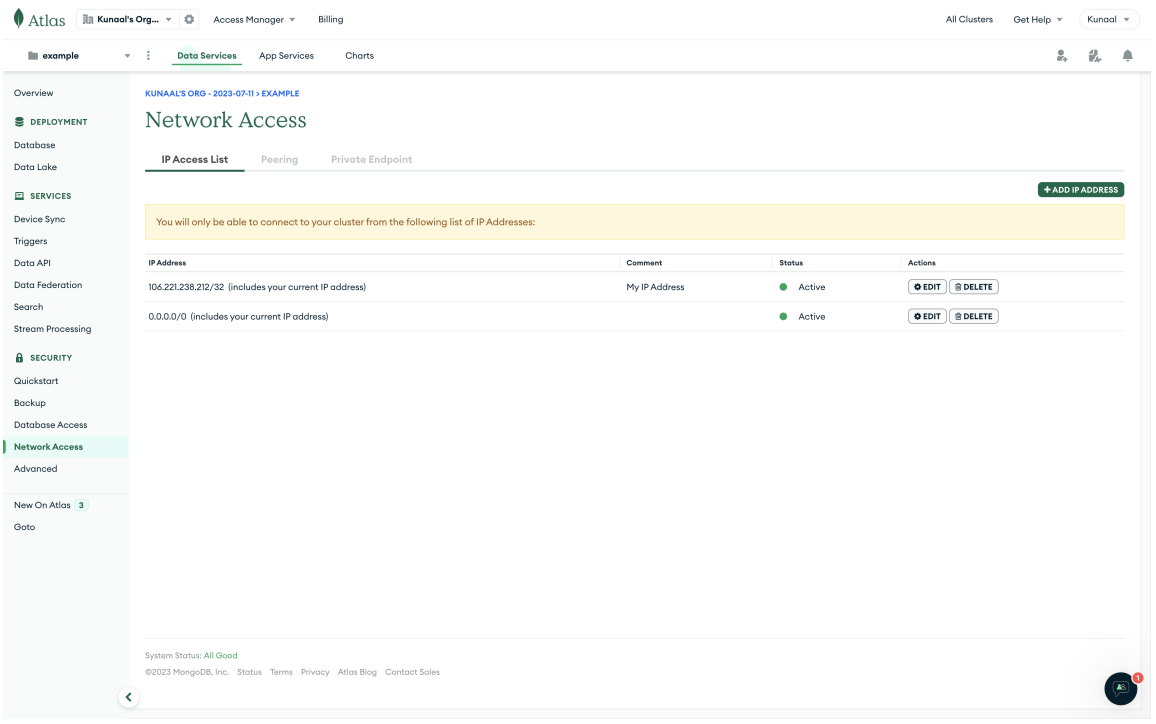
37. Click on “Add IP address”



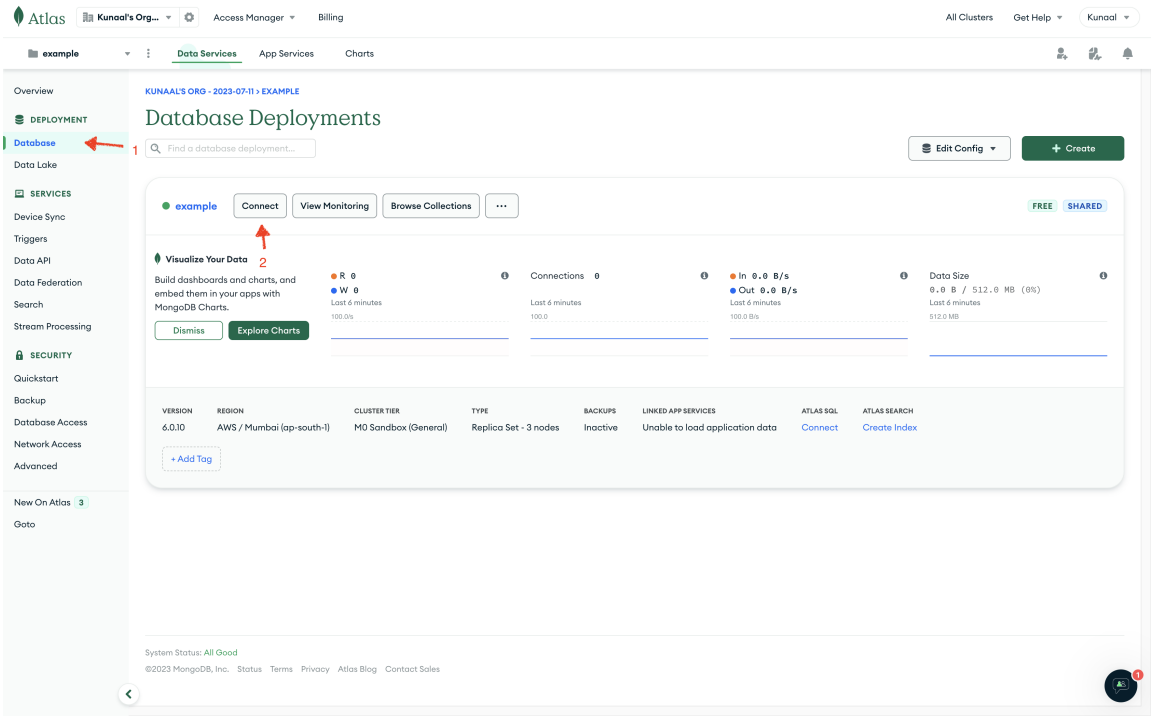
38. Then click on “Allow access from anywhere” & confirm.



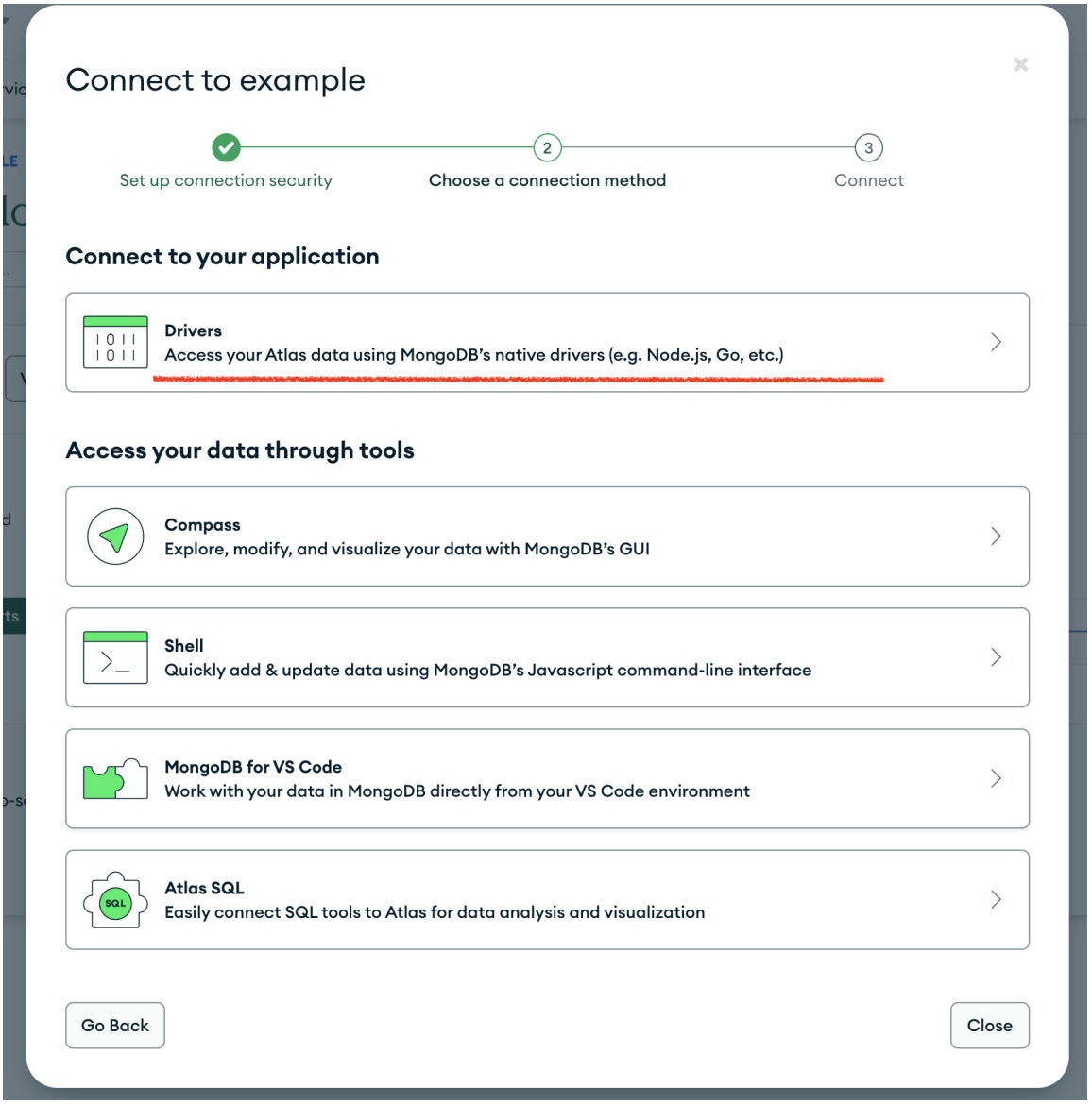
39. After that it will take some time and you will then see this in your network list. (Now you can access the database from any IP)



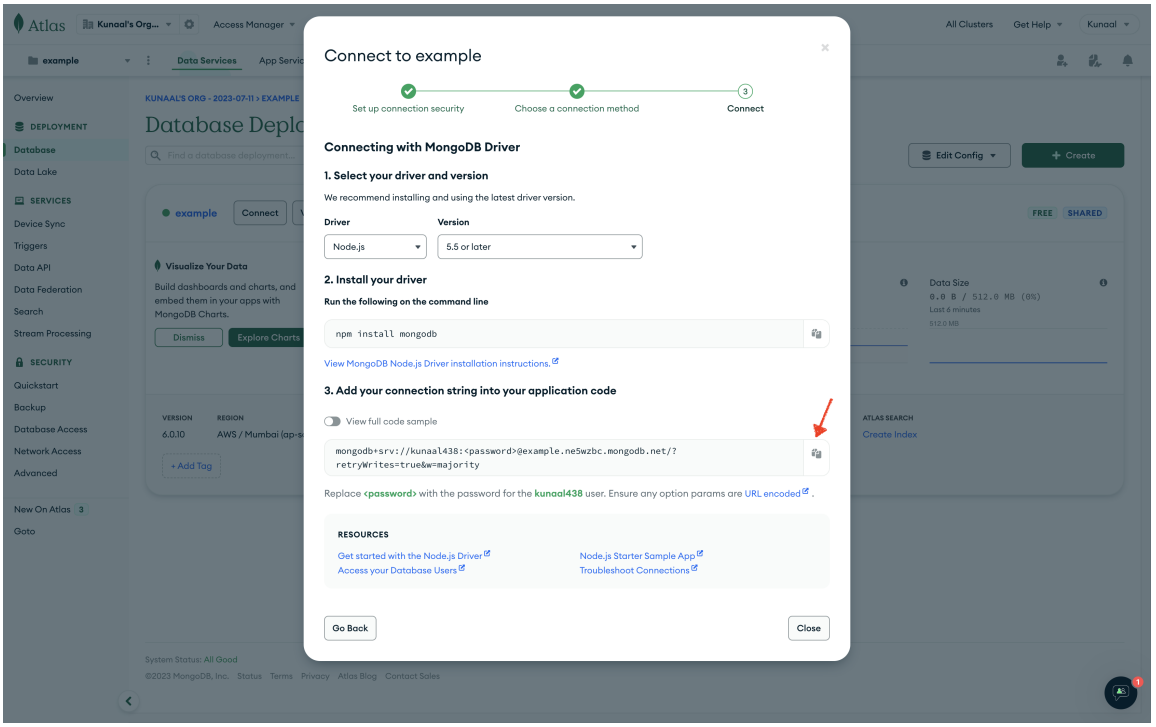
40. Now go to “database” under “deployment” & click on “connect”



41. Now click on “drivers”



42. Copy the database URL



43. Paste this URL in ".env" file inside "server" folder after "DB_LOCATION="

```
DB_LOCATION=mongodb+srv://kunaal438:<password>@example.ne5wzbc.mongodb.net/?retryWrites=true&w=majority
```

44. Now replace the "<password>" from the URL with your user's password that you have generated on step 35.

```
DB_LOCATION=mongodb+srv://kunaal438:sEagsRx8C0PDM0hX@example.ne5wzbc.mongodb.net/?retryWrites=true&w=majority
```

45. Now, let's create a secret key for our JWT. So in your terminal. Run `node`
46. It will start node inside terminal. So just copy below command and paste it inside terminal and run it.

    cmd - require('crypto').randomBytes(64).toString('hex')

47. This will give you a long string.

```
> require('crypto').randomBytes(64).toString('hex')
'95bcdd7d810cbe50a7b4e0772191e22d720f465c3e8310c52fb46f419ed8878000058dedded5f97aab77fb710801171d97a192240d7e50fb2e851b6cf694d20b'
```

48. Copy this string and paste it inside ".env" of "server" folder after "SECRET_ACCESS_KEY="

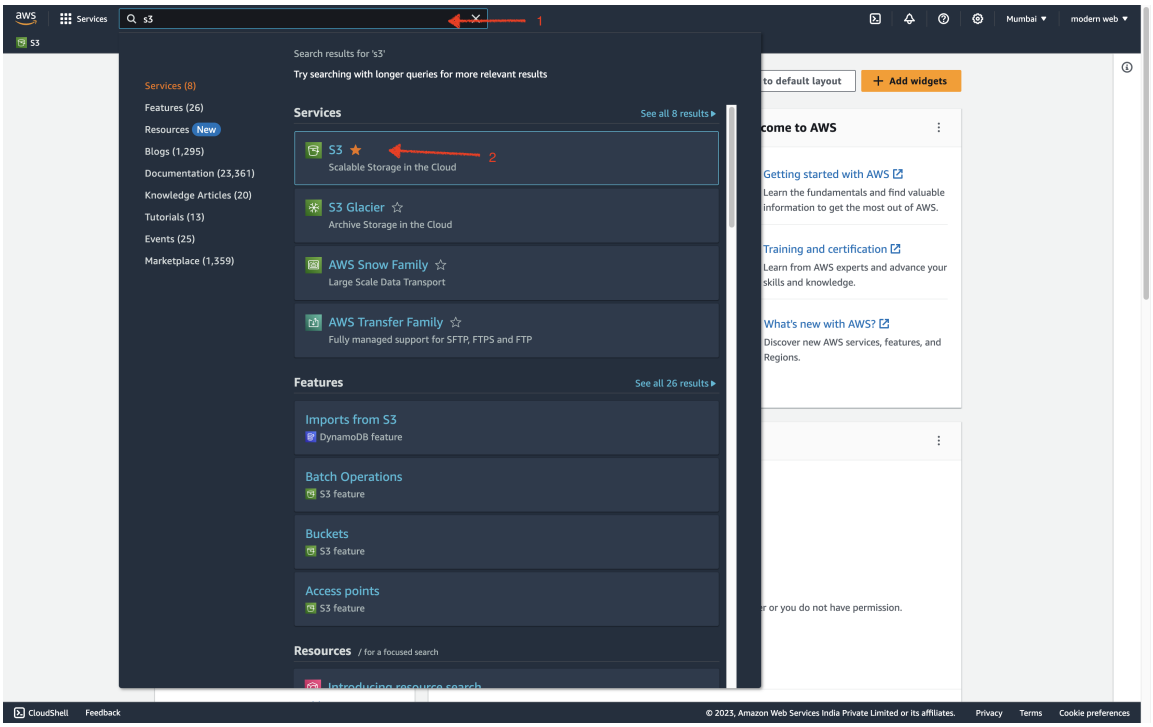
```
SECRET_ACCESS_KEY=95bcdd7d810cbe50a7b4e0772191e22d720f465c3e8310c52fb46f419ed8878000058dedded5f97aab77fb710801171d97
```

49. Now let's setup AWS. Go to - https://aws.amazon.com/

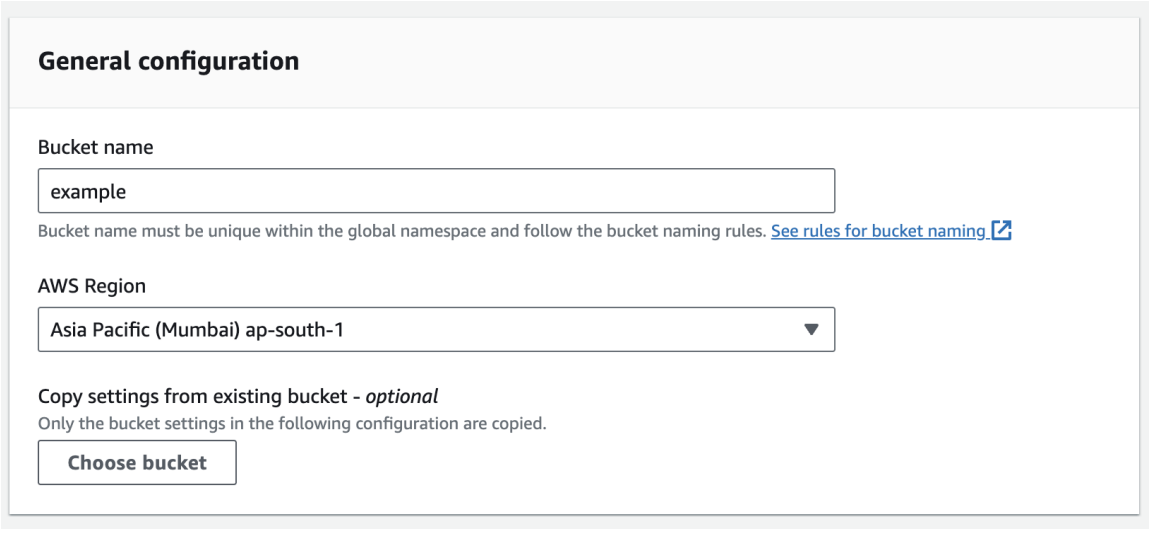
50. Now either sign in to your AWS account or create your account there.

    Note - AWS require your credit card info to create an account. Don’t worry AWS is free, it will only charge you when you reach the free limit of it and since this website is just for your portfolio not a real world project, you won’t exceed the free limit.

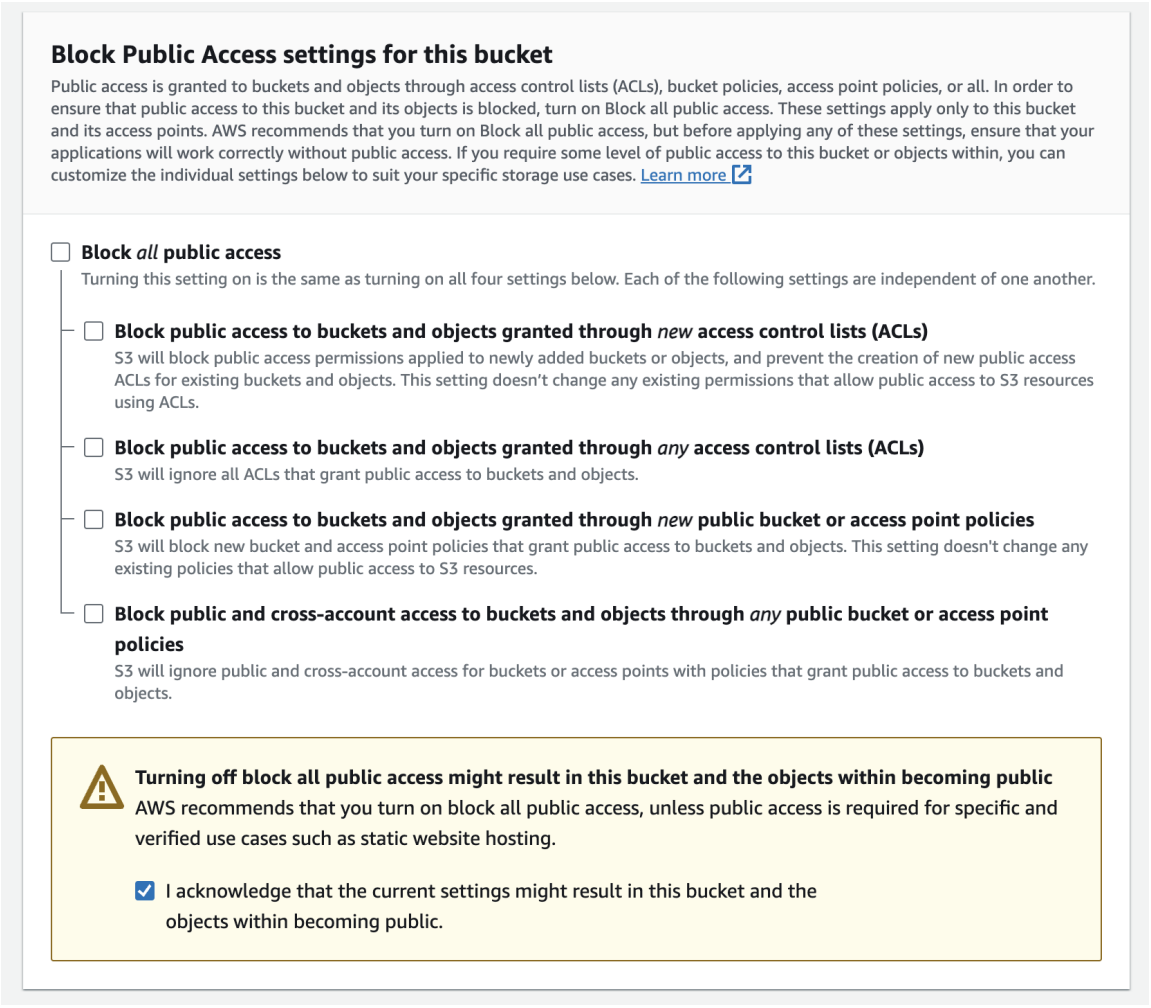
51. Now search for S3



52. Now click on “create bucket” a yellow button on right side.
53. Give a “bucket name”
54. Choose a region nearest to you.

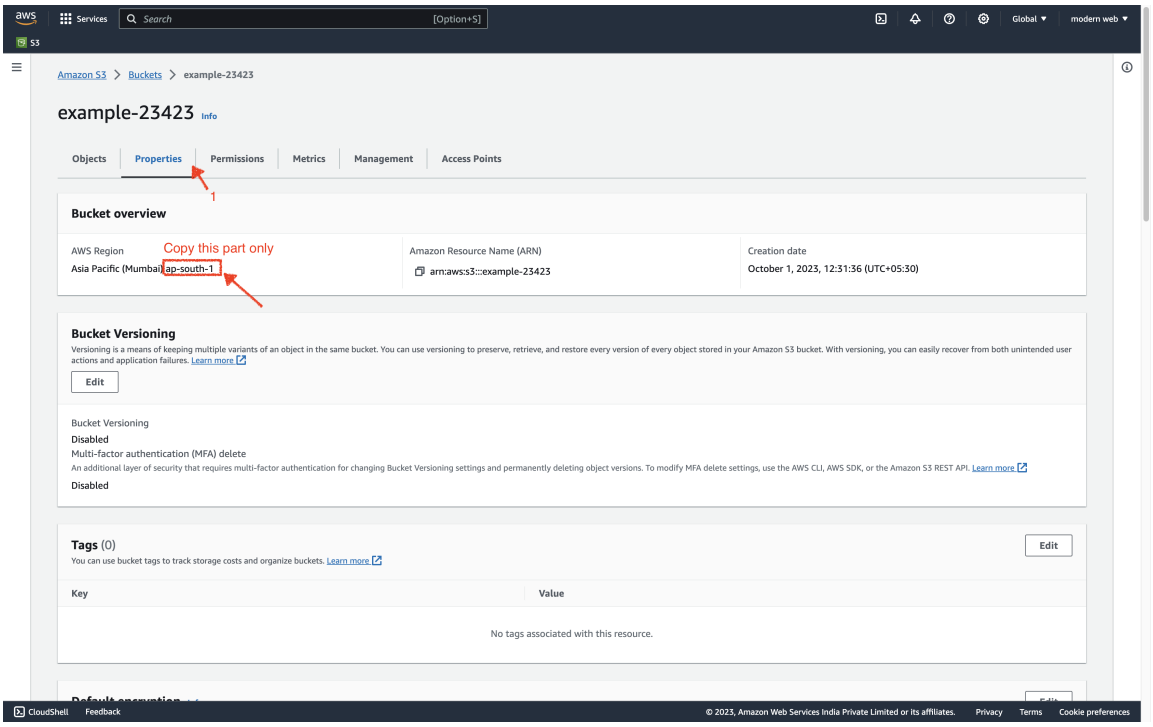


55. Uncheck “block all public access”



56. And no need to change anything. Just create the bucket by clicking on the button below.

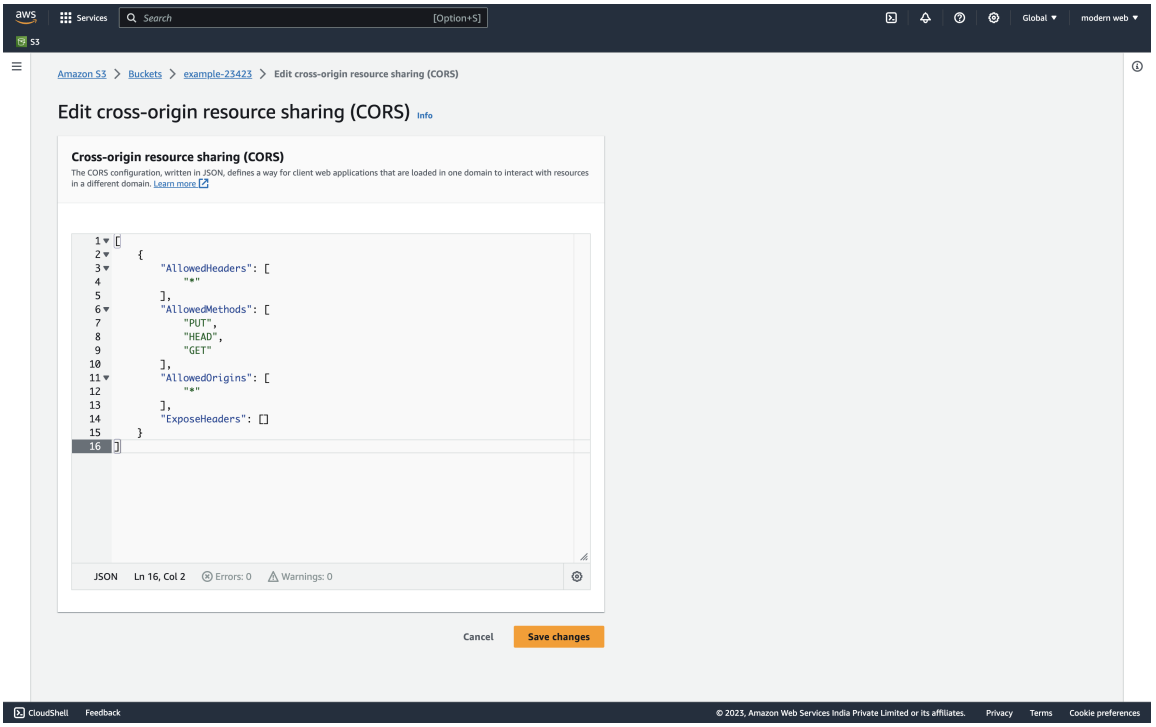
57. You will get redirected to all bucket's page. Click on the bucket name that you just create to open it.
58. Now go to projects and copy the region.



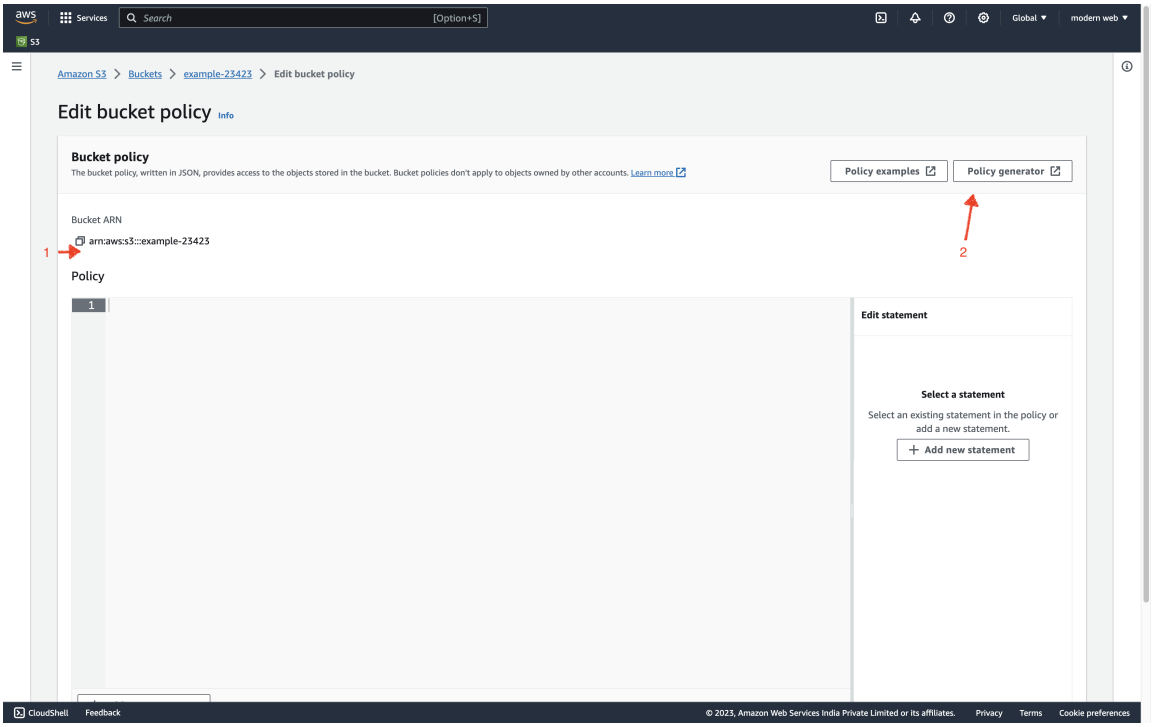
59. And paste this inside ".env" of "server" folder after "BUCKET_REGION="

```
9   BUCKET_REGION=ap-south-1
```

60. Now go to "permissions" tab. (its right after "properties" tab)
61. Scroll down and find "**Cross-origin resource sharing (CORS)"** at the end. Click on edit.
62. Now open the "cors.txt" file that you have with 2 folders in the source code. Whatever written in that file, copy that and paste it inside AWS CORS.

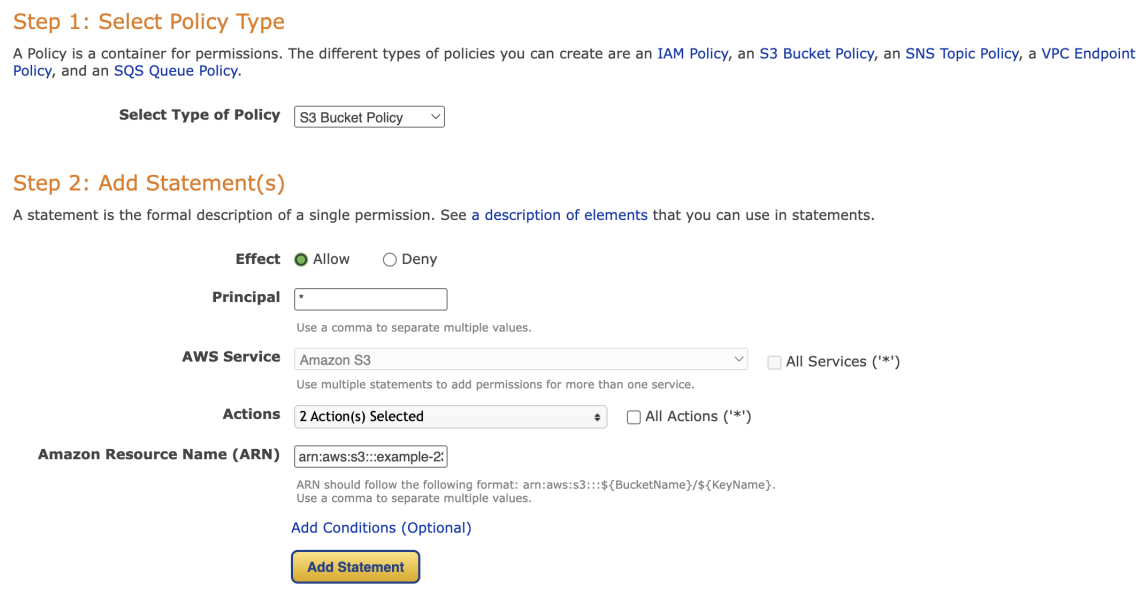


63. Click on save changes. This will allow AWS to accept request from all IP.
64. Now scroll to top and find “Bucket Policy”, click on Edit.
65. Copy the “arn” first and then go to “policy generator”

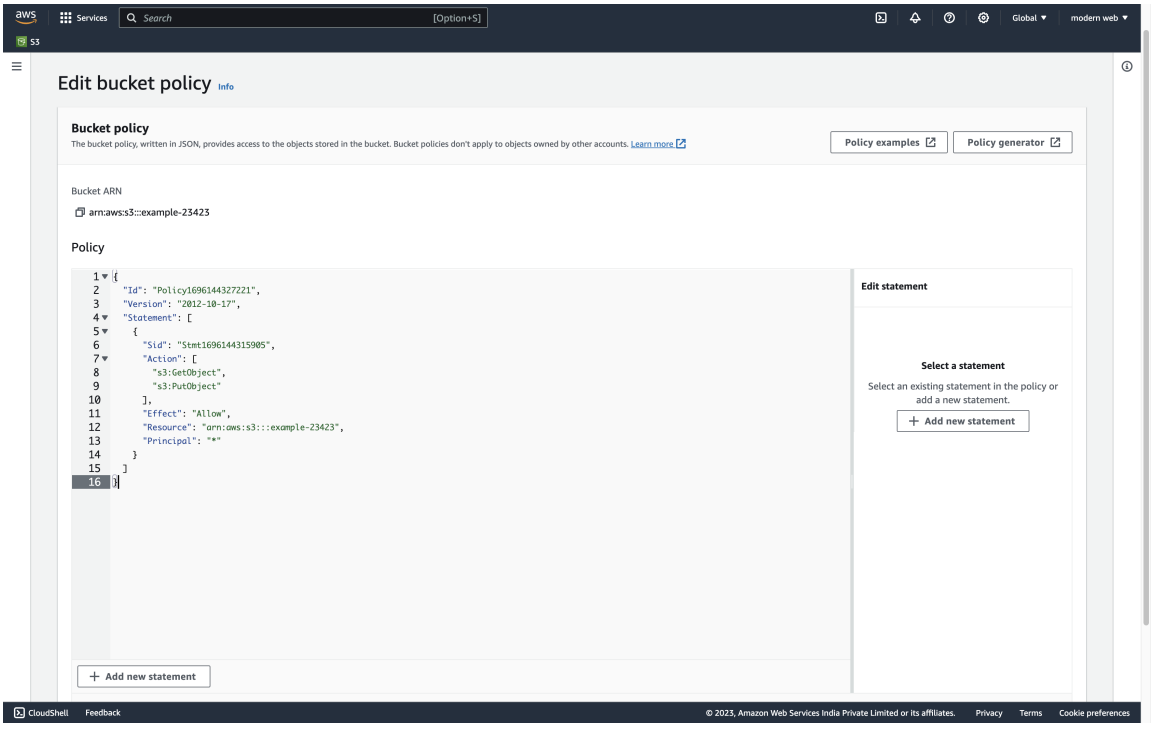


66. Select “S3 Bucket Policy” in “**Select Type of Policy”**
67. Type * in “**Principal”**

68. In “**Actions”**, Select “GetObject” and “PutObject”
69. Now paste the “arn” you copied on step 65 to ARN field.
70. Then click on “add statement”



71. Click on “generate policy” & copy the text its giving you.
72. Paste those text inside the “Bucket Policy” editor

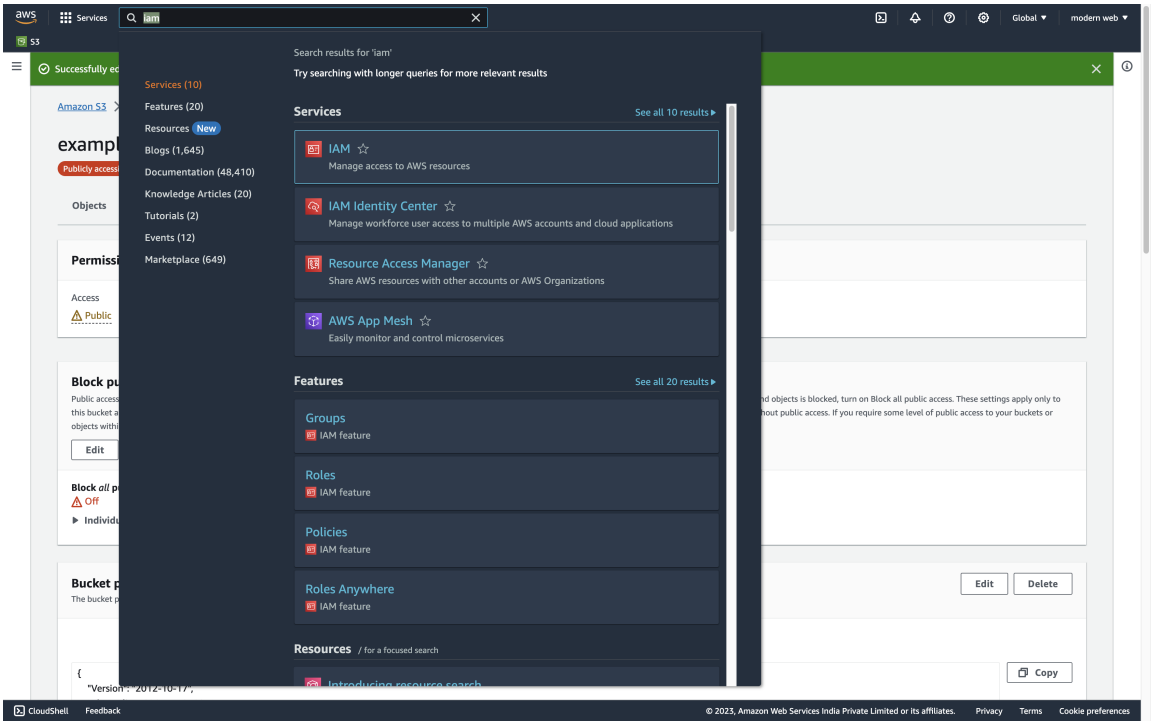


73. Now add “ /* “ at the end of “Resources” of the policy.

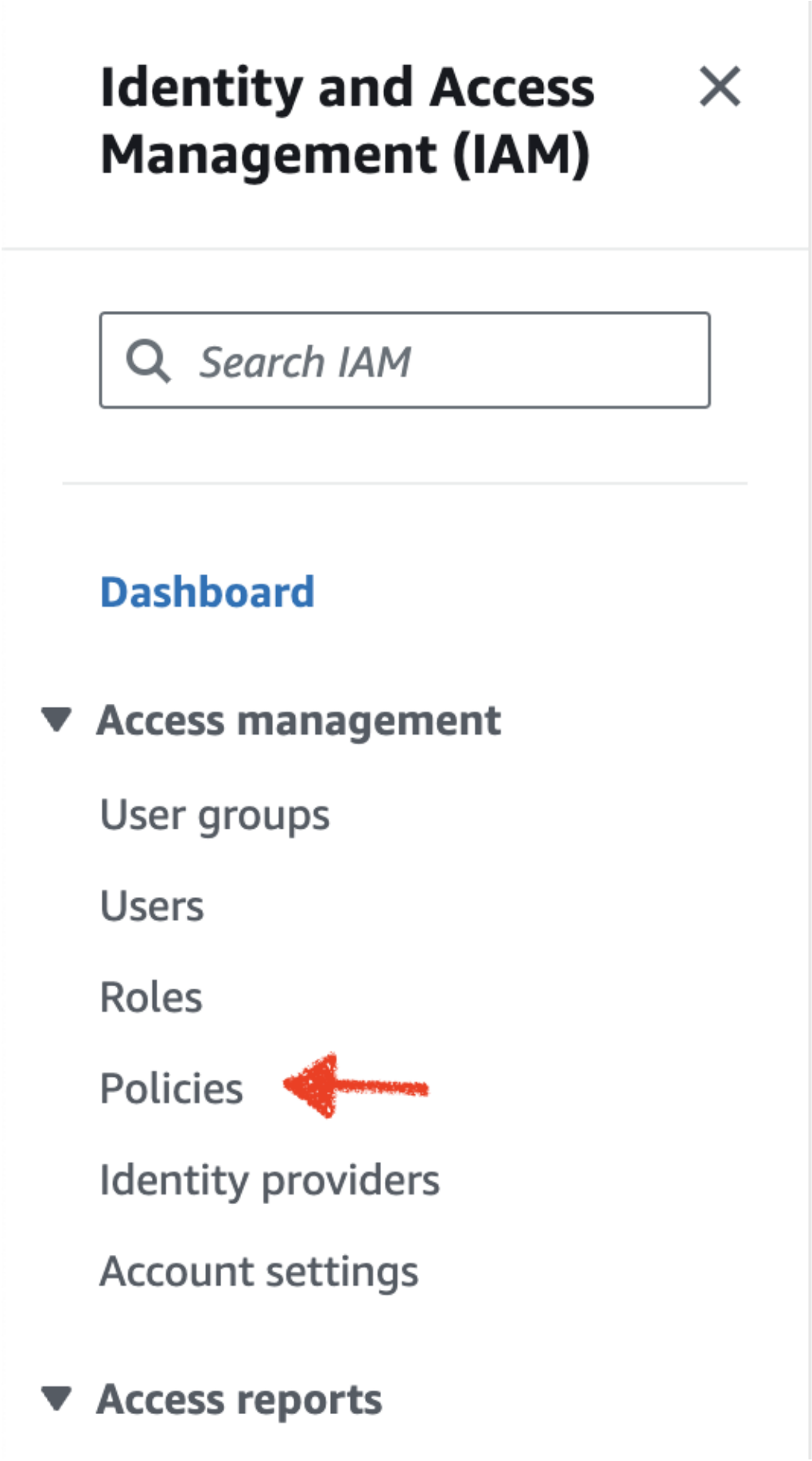
```
12          "Resource": "arn:aws:s3:::example-23423/*",
```

74. Click on “save changes” now to save the policy.

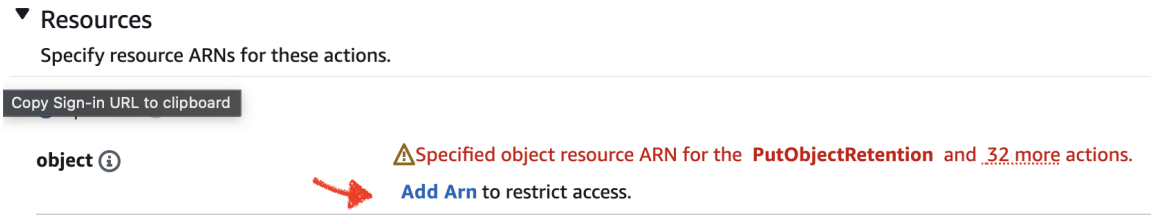
75. All the Bucket setting is done. Now we need keys to connect our server with AWS. So go to search and search for “IAM”



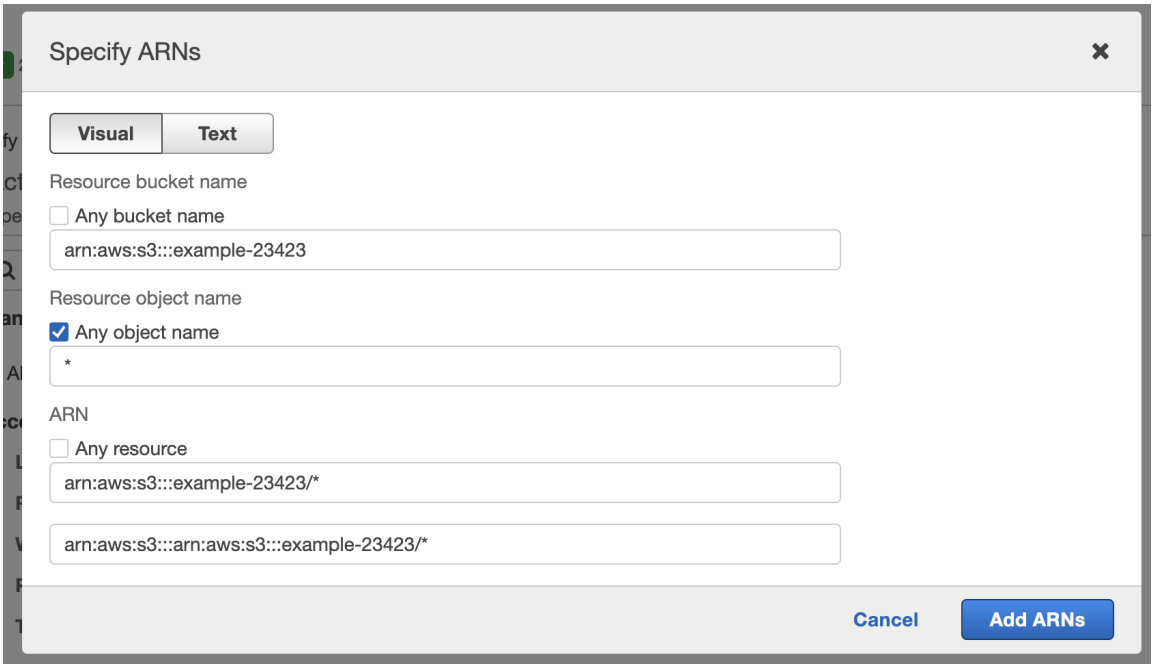
76. Now go to “policies”


Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports

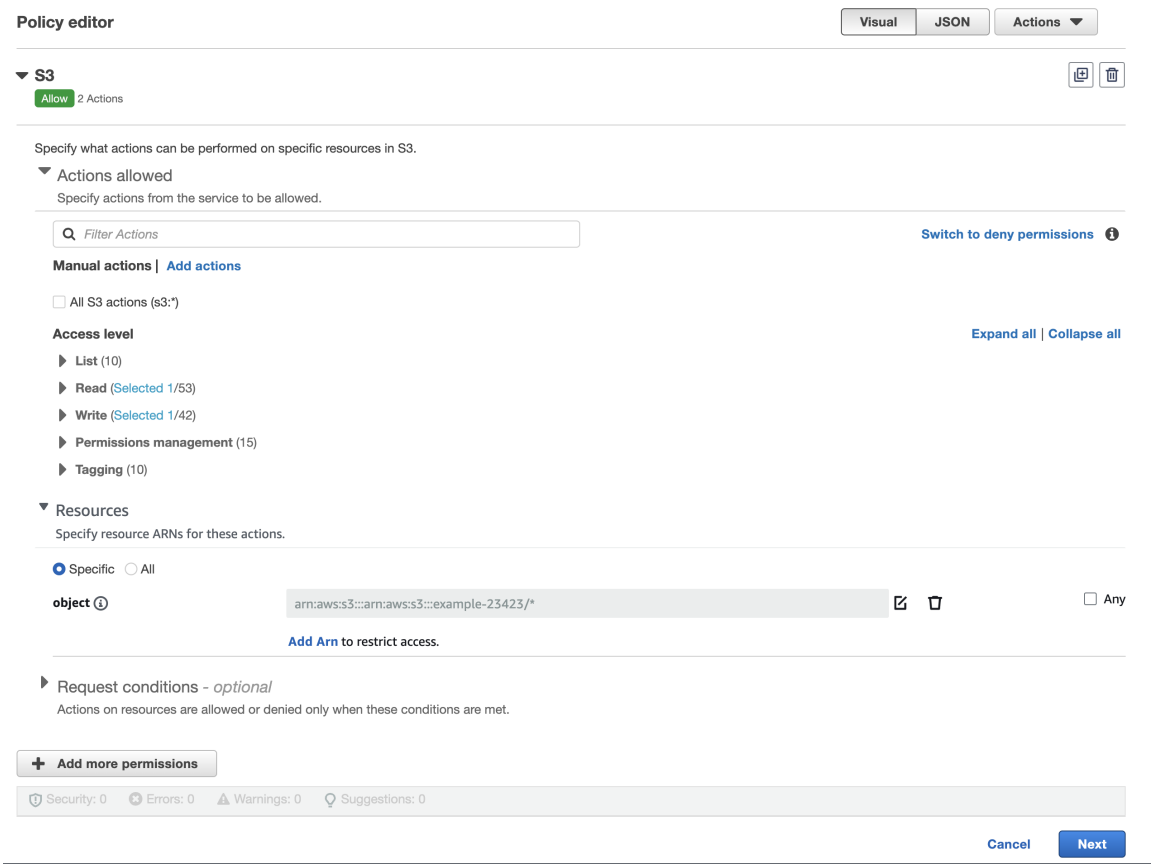
77. Click on create policy. (a blue color button on the right side)
78. Under “select services” select S3. (You can search S3 there and click it)
79. Under “Actions allowed”. Search for “GetObject” & “PutObject” and check both of them.
80. Under Resources, choose “specific” and click on “Add Arn”



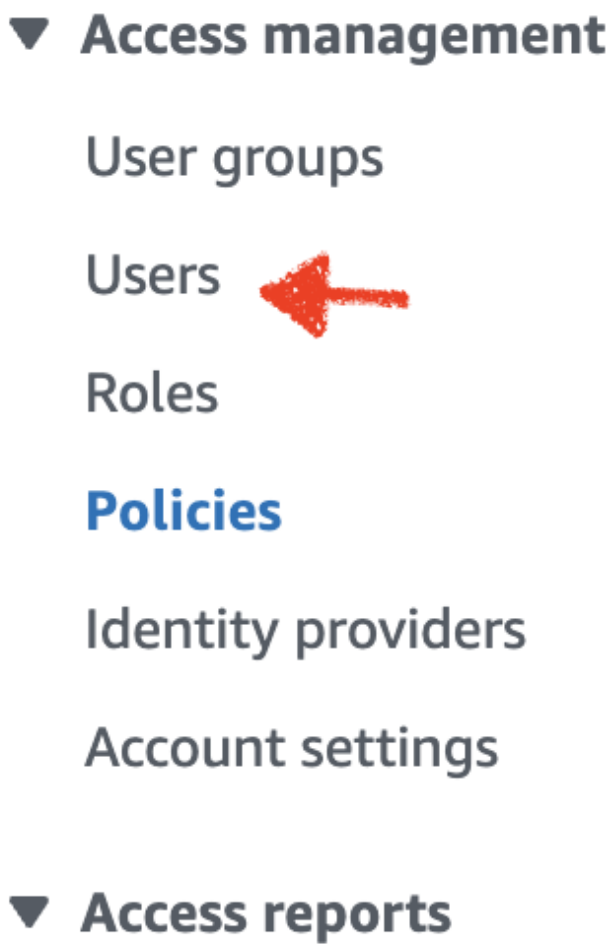
81. And paste your bucket “arn” here. the same that you used to generate the policy of bucket.



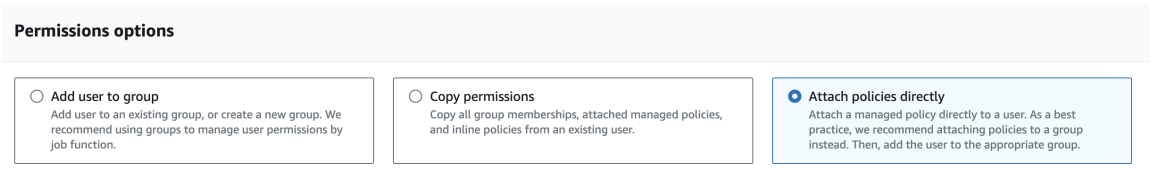
82. Make sure everything else looks same like me (except the arn of your bucket) . Click on “Add Arns” now.
83. Now click “next”



84. Give any name to the policy. To make things easy give your bucket name to the policy as well, so it will be easy for you to tell this policy is for what bucket.
85. And Click on “create policy”
86. Now go to “users” from the sidebar.

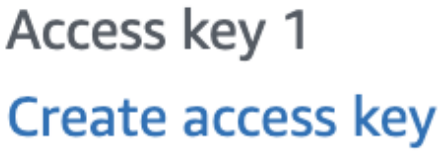


87. Click on “create user” (button on the right side of the page)
88. Give any user name. Again to make things easy give your bucket name here, so it will be easy for you to tell this user can access this bucket.
89. In “**Permissions options”** Select “Attach policies directly”

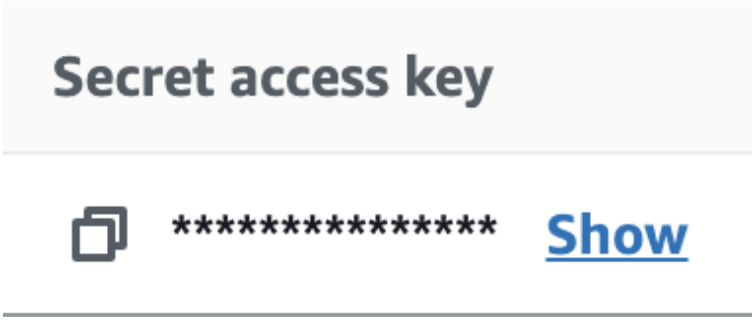


90. Now search for the policy name that you just generated on step 85 and check that.
91. Click “Next”.
92. Then click on “Create User”.
93. It will create the user.

94. Now, you will get redirected to the page where you will get all the list of users. Search the user name that you just created and click on it to open it.
95. Under “Summary”. Click on “Create access key”



96. Under use cases, select “other” and click “next”
97. Skip the tag part and click on “create access key”
98. You will get 2 keys. Now do not click on “Done” because you will get access to “secret key” only once so better copy that first.
99. Copy “secret key”



00. Now again go to “.env” of “server” folder and paste it after “AWS_SECRET_ACCESS_KEY=” (with no space)
01. After that come back to AWS and copy “**Access key”** and paste it after “AWS_ACCESS_KEY=” inside “.env” file.

    Note - I can’t show you the keys so therefore no screenshots here but just make sure you add no space after = for both variables and save the file.

02. Now everything is connected. You can start your server now.
03. Just one more this we have to do. Open “.env” file of “blogging website” folder. There you will find “VITE_SERVER_DOMAIN” variable.

04. Add your server URL there otherwise the frontend wont be able to make request to the server.

```
VITE_SERVER_DOMAIN=http://localhost:3000
```

Note - make sure you don't add "/" at the end of the localhost URL otherwise it wont work. Also if you server run on some different port rather that 3000 change the 3000 to that port number as well.

05. Now you can start you frontend and backend code on sever and access the website on browser.
06. To run frontend, open "blogging website" folder in terminal and run `npm run dev` cmd it will start the vite server for you to access the website on localhost.
07. To run the server, open "server" folder in terminal and run `npm start` cmd, it will start the node server on localhost with the PORT number 3000. mentioned in the "server.js" file.

    Note - If you get error like "PORT is already in use then in server.js file on line 20 change the PORT to any other number instead of 3000 to start you server on different PORT. Change the PORT number until you find a not running PORT. generally if you just change PORT to 3001 or 3002 this should work.

    Note - Also if you change the PORT in server.js it will start the server on different PORT, so make sure you change the PORT number inside ".env" of "blogging website" where the VITE_SERVER_DOMAIN is localhost. change that URL's PORT to your current server PORT.

Now you can access your website and try using it. Adding data to it, building more features on top of that, its up to you what you want to do with that. To access website every time you have start your "vite" server and "node" server. Mean follow the step 106 and 107 every time you want to access the website.