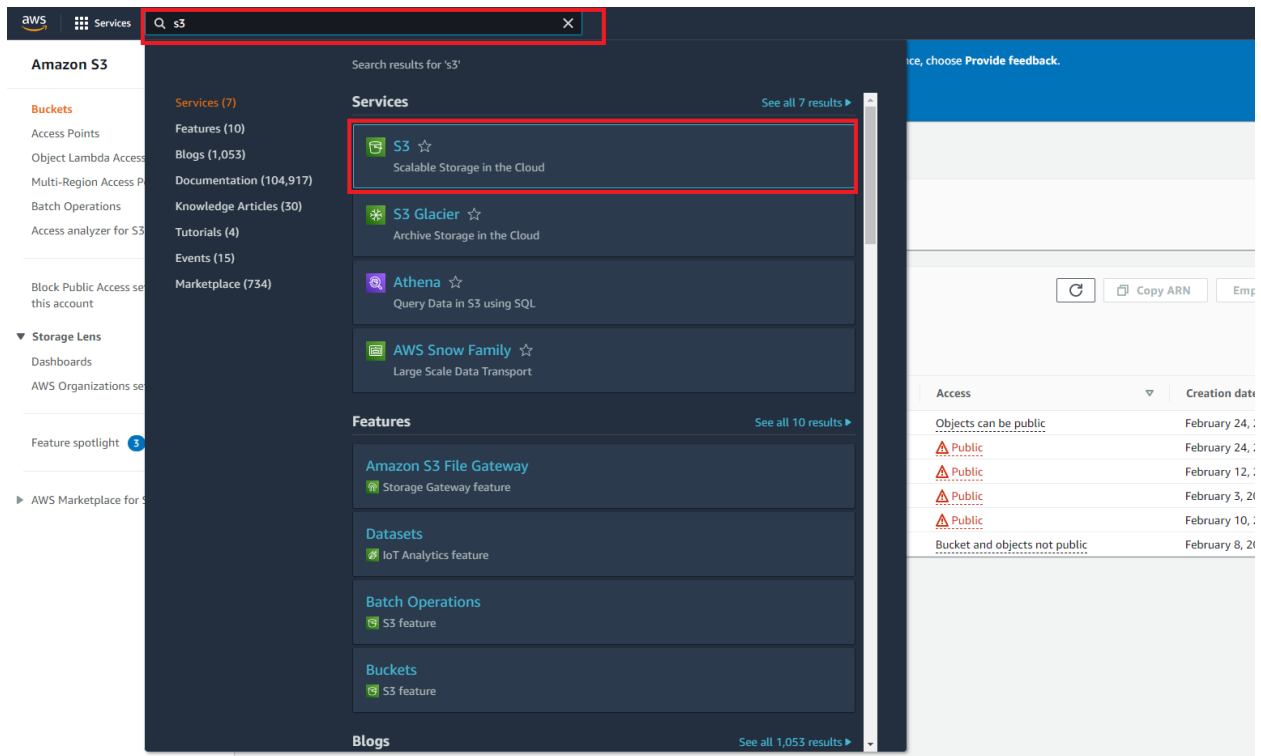
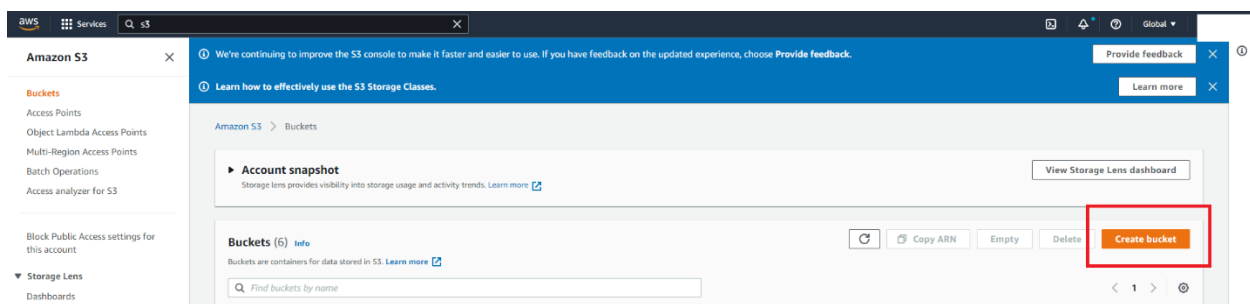


1. Recreate the database by running the below commands:  
drop database if exists online\_store\_db;  
create database online\_store\_db;
2. Clone the project from GitHub:  
git clone [git@github.com:achiever102/onlineshop.git](https://github.com/achiever102/onlineshop.git)
3. Create an AWS S3 bucket to store application images:
  - Login to AWS
  - Go to S3 service



- Click on Create Bucket:



- Enter bucket name and make the bucket public:

Provide feedback

✕

Amazon S3 > Buckets > Create bucket

Create bucket [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

General configuration

Bucket name

appbucket12029883

Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

AWS Region

US East (N. Virginia) us-east-1

Copy settings from existing bucket - optional

Only the bucket settings in the following configuration are copied.

Choose bucket

Object Ownership [Info](#)

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

☒ ACLs disabled (recommended)  
All objects in this bucket are owned by this account.  
Access to this bucket and its objects is specified using only policies.

☐ ACLs enabled  
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership

Bucket owner enforced

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback, choose [Provide feedback](#).

Bucket owner enforced

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or its objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☐ Block all public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

☐ Block public access to buckets and objects granted through new access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

☐ Block public access to buckets and objects granted through any access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

☐ Block public access to buckets and objects granted through new public bucket or access point policies

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

☐ Block public and cross-account access to buckets and objects through any public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Turning off block all public access might result in this bucket and the objects within becoming public. AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

☒ I acknowledge that the current settings might result in this bucket and the objects within becoming public.

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose [Provide feedback](#).

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

☒ Disable  
☐ Enable

Tags (0) - optional

Track storage cost or other criteria by tagging your bucket. [Learn more](#)

No tags associated with this bucket.

Add tag

Default encryption

Automatically encrypt new objects stored in this bucket. [Learn more](#)

Server-side encryption

☒ Disable  
☐ Enable

► Advanced settings

After creating the bucket you can upload files and folders to the bucket, and configure additional bucket settings.

Cancel

Create bucket

- Click on the created bucket:

The screenshot shows the Amazon S3 console interface. On the left is a navigation sidebar with options like Buckets, Access Points, and Storage Lens. The main content area displays a list of buckets. A table with columns 'Name', 'AWS Region', 'Access', and 'Creation date' is shown. The first row, representing the bucket 'appbucket12029883' in the 'US East (N. Virginia) us-east-1' region with 'Objects can be public' access, is highlighted with a red rectangular box.

- Click on Permissions:

This screenshot shows the 'Permissions' tab selected for the bucket 'appbucket12029883'. The 'Permissions overview' section indicates 'Access: Objects can be public'. Below this, the 'Block public access (bucket settings)' section shows 'Block all public access' is currently 'Off' with a warning icon. A red box highlights the 'Permissions' tab in the sub-navigation bar at the top of the content area.

- Scroll down to bucket policy and click on edit:

The screenshot displays the 'Bucket policy' section. It shows a message 'No policy to display.' and an 'Edit' button. A red box highlights the 'Edit' button, which is used to modify the bucket's policy.

- Copy the bucket ARN:

Amazon S3

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose [Provide feedback](#).

Amazon S3 > Buckets > appbucket12029883 > Edit bucket policy

### Edit bucket policy

**Bucket policy**  
The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

**Bucket ARN**  
arn:aws:s3::appbucket12029883

**Policy**

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "Statement1",
6       "Principal": {},
7       "Effect": "Allow",
8       "Action": [],
9       "Resource": []
10    }
11  ]
12 }
```

**Edit statement Statement1** Remove

**1. Add actions**  
Choose a service

**Available**  
AMP  
API Gateway  
API Gateway V2  
Access Analyzer  
Account  
Activate  
Alexa for Business  
Amplify  
Amplify Admin  
Amplify UI Builder

- Replace the existing policy with the below and make sure to replace the highlighted ARN with yours:

Amazon S3

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose [Provide feedback](#).

Amazon S3 > Buckets > appbucket12029883 > Edit bucket policy

### Edit bucket policy

**Bucket policy**  
The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

**Bucket ARN**  
arn:aws:s3::appbucket12029883

**Policy**

```
1 {
2   "Id": "Policy1648556313478",
3   "Version": "2012-10-17",
4   "Statement": [
5     {
6       "Sid": "Stmt1648556313474",
7       "Action": "s3:*",
8       "Effect": "Allow",
9       "Resource": "arn:aws:s3::appbucket12029883/*",
10      "Principal": ""
11    }
12  ]
13 }
```

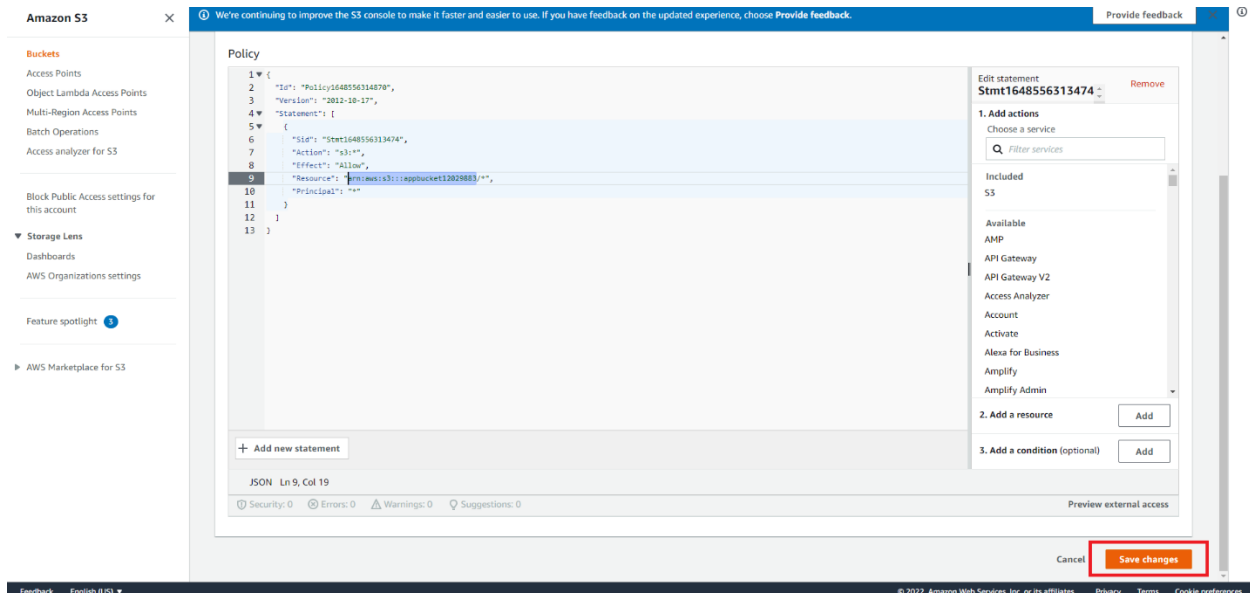
**Edit statement Stmt1648556313474** Remove

**1. Add actions**  
Choose a service

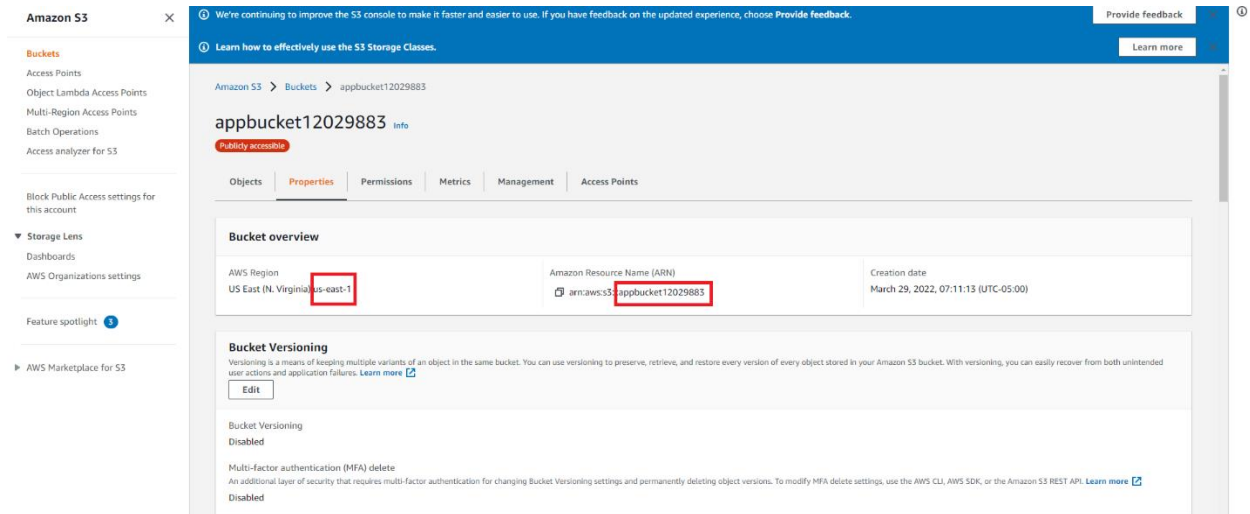
**Included**  
S3

**Available**  
AMP  
API Gateway  
API Gateway V2  
Access Analyzer  
Account  
Activate  
Alexa for Business

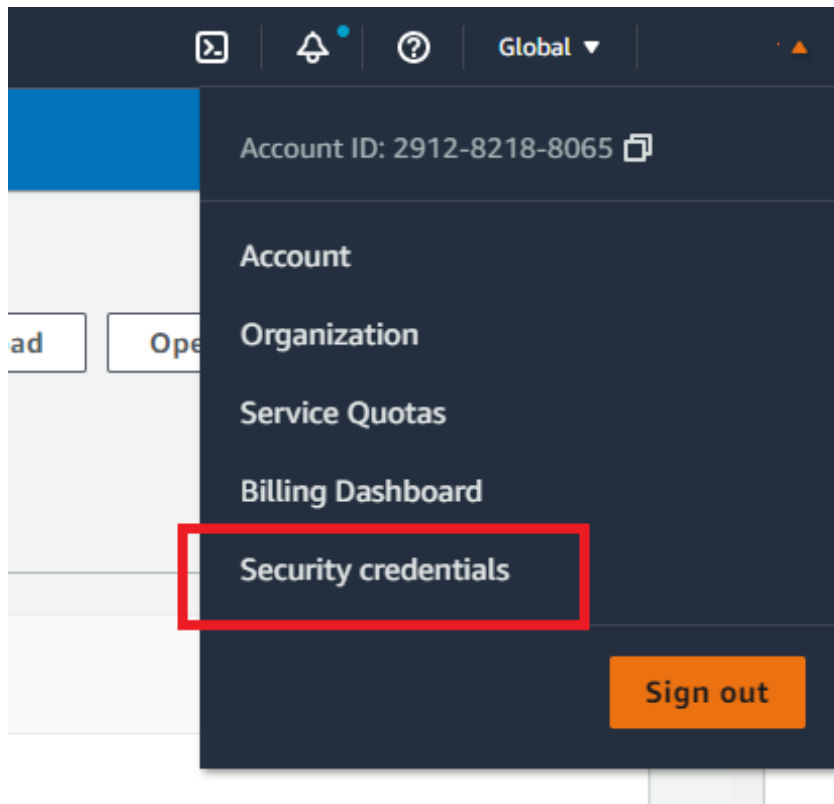
- Click on save changes:



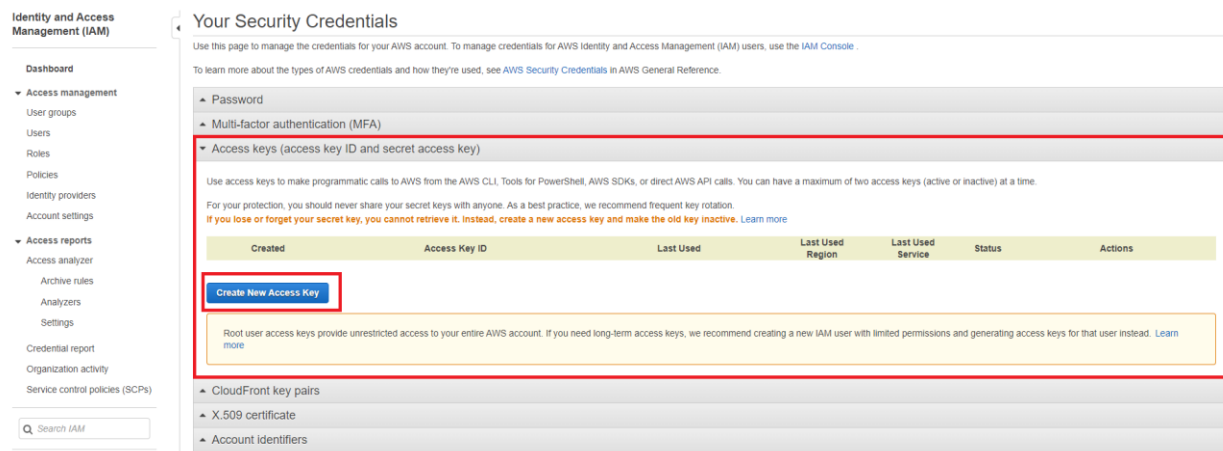
- Go to bucket properties and copy the bucket region and bucket name:



- Create AWS access key:
  - Click on the user's drop down list and click on Security Credentials:

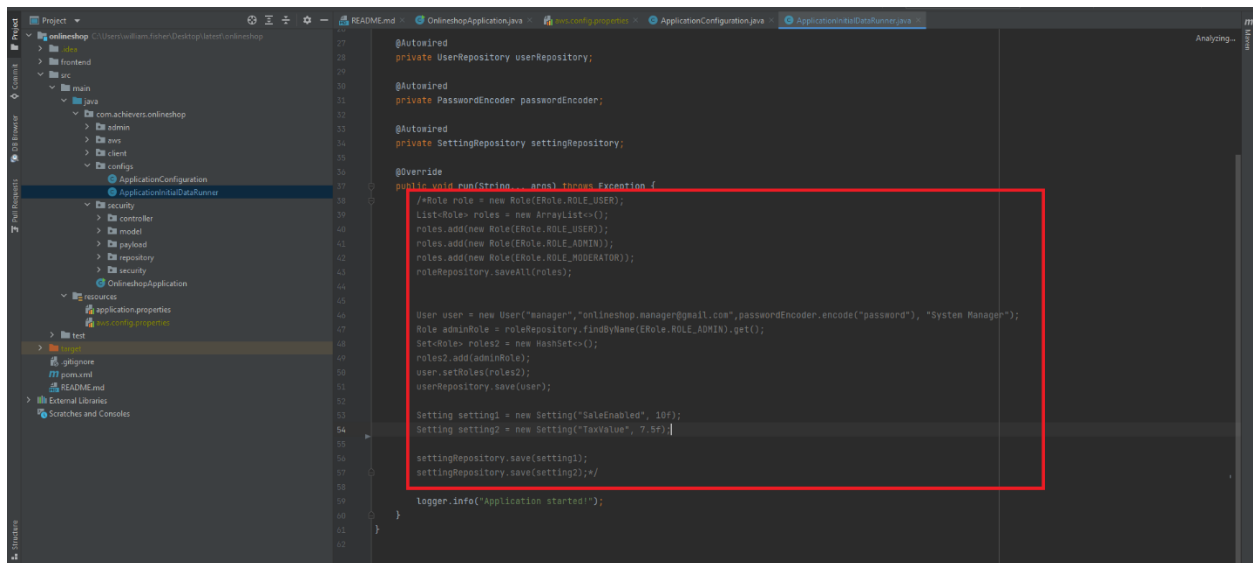


- Click on create new access key:



- Open the downloaded CSV file and copy the access key and secret key.

4. Open the backend project in IntelliJ and create a file with name `aws.config.properties` in the directory: `src/main/resources` then add the below parameters to the file:  
`accesskey=access key from the downloaded file`  
`secretKey=secret key from the downloaded file`  
`bucketName=bucket name`  
`bucketUrl=https://bucket-name.s3.region.amazonaws.com/`  
`region=region`
5. To insert the application initial data into the database (manager user username and password in addition to ACLs), uncomment the below lines in the backend class `ApplicationInitialDataRunner.java`



```
17  @Autowired
18  private UserRepository userRepository;
19
20  @Autowired
21  private PasswordEncoder passwordEncoder;
22
23  @Autowired
24  private SettingRepository settingRepository;
25
26  @Override
27  public void run(String... args) throws Exception {
28      //Role role = new Role(ERole.ROLE_USER);
29      List<Role> roles = new ArrayList<>();
30      roles.add(new Role(ERole.ROLE_USER));
31      roles.add(new Role(ERole.ROLE_ADMIN));
32      roles.add(new Role(ERole.ROLE_MODERATOR));
33      roleRepository.saveAll(roles);
34
35      User user = new User("manager", "onlineshop.manager@gmail.com", passwordEncoder.encode("password"), "System Manager");
36      Role adminRole = roleRepository.findByName(ERole.ROLE_ADMIN).get();
37      Set<Role> roles2 = new HashSet<>();
38      roles2.add(adminRole);
39      user.setRoles(roles2);
40      userRepository.save(user);
41
42      Setting setting1 = new Setting("SaleEnabled", 10f);
43      Setting setting2 = new Setting("TaxValue", 7.5f);
44
45      settingRepository.save(setting1);
46      settingRepository.save(setting2);
47
48      Logger.info("Application started!");
49  }
```

6. Download and install JDK 11
7. Start the backend project in IntelliJ
8. Comment the same lines out. You just need to run this script once.
9. Start the frontend project in VSCode and run the commands:
  - `npm install`
  - `npm start`