**4)Create an s3 bucket and deploy the reactjs static application on it and provide with the public url for use.**

**Step1:** Create an S3 bucket in the AWS Management Console. Choose a globally unique bucket name, and configure the bucket settings like region, access permissions, and logging. Make sure to note down the name of our bucket.

**Step2:** Build our ReactJS application using a command like npm run build. This will create a build directory containing your application's static files

**Step3:** Upload the contents of our build directory to our S3 bucket using the AWS Management Console or any S3 client tool like AWS CLI.

**Step4:** we can do this by navigating to the "Properties" tab of our bucket and selecting "Static website hosting". Set the "Index document" to the entry point of our application, which is typically index.html.

**Step5:** Make our bucket contents publicly accessible by adding a bucket policy that allows read access to all users. We can do this by navigating to the "Permissions" tab of our bucket and selecting "Bucket policy".

Use the following policy

**{**

**"Version": "2012-10-17",**

**"Statement": [**

**{**

**"Sid": "PublicReadGetObject",**

**"Effect": "Allow",**

**"Principal": "\*",**

**"Action": [**

**"s3:GetObject"**

**],**

**"Resource": [**

**"arn:aws:s3:::<bucket-name>/\*"**

**]**

**}**

**]**

**}**

Replace <bucket-name> with the name of our S3 bucket.

Once we have configured our S3 bucket, we can access our ReactJS application using the public URL of our S3 bucket, which will be in the format http://<bucket-name>.s3-website-<region>.amazonaws.com.