

[Open in app](#)

★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



Run AWS on Your Laptop. Introduction to LocalStack.



Samsonk · [Follow](#)

Published in [AWS Tip](#)

5 min read · Aug 18, 2023



Listen



Share



More



Cover Image

What is LocalStack

LocalStack is a cloud service emulator that runs AWS services solely on your laptop without connecting to a remote cloud provider .

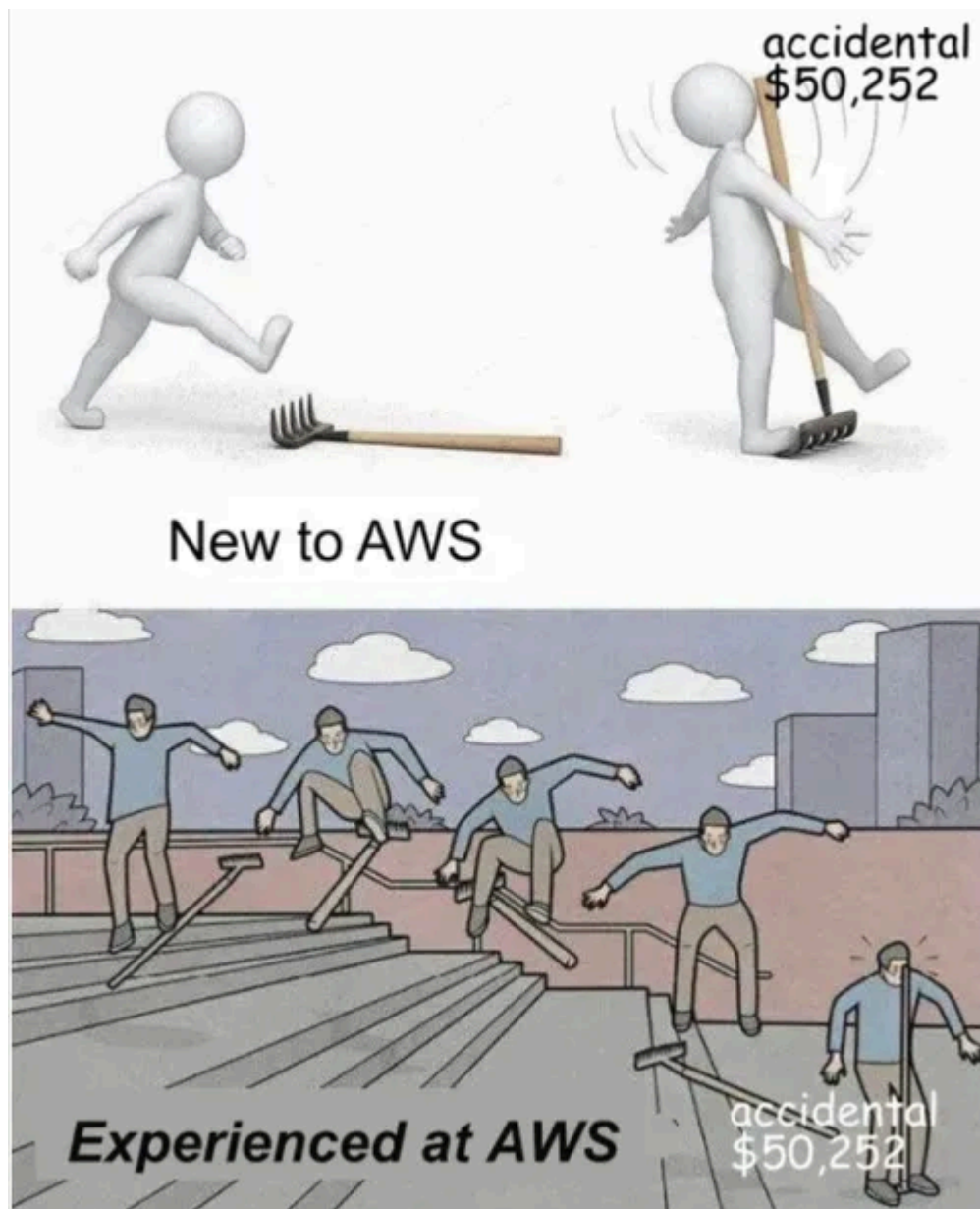
How can it help us as a developer?

Whether you are,

1. A beginner who is looking to practice your AWS skills but don't have access to credit card which is required upon AWS registration.

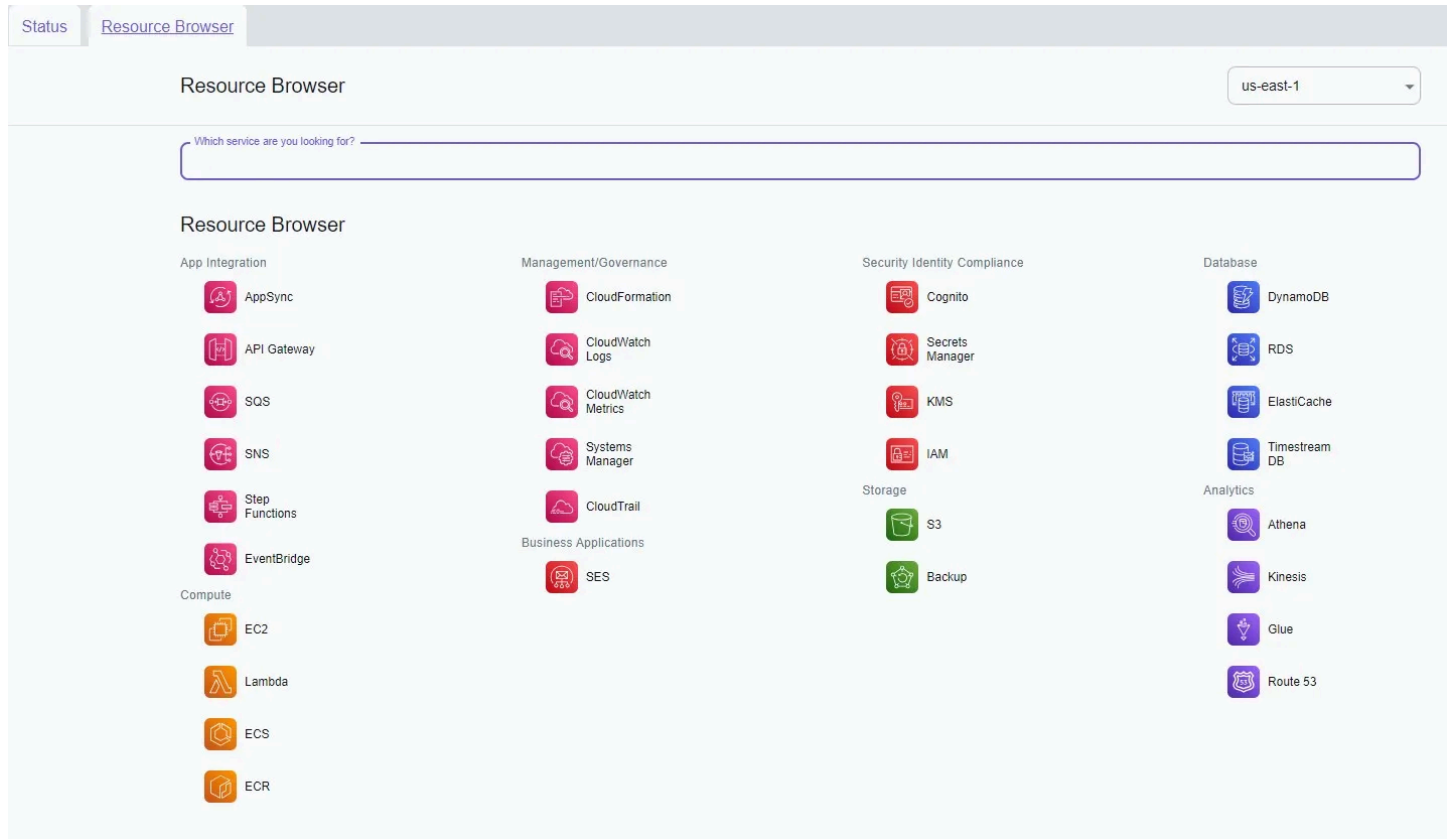
2. A student, who wants to gain hands-on experience with AWS services without spending any money.
3. A professional who wants to troubleshoot or test your infrastructure configurations offline on your machine, without setting up a separate cloud environment for testing, and then seamlessly transition to your main AWS production environment once everything is ready and optimized,

then LocalStack has got you covered. Because the last thing you want to do is set up an AWS environment to improve your skills, only to accidentally shoot yourself in the foot with a ridiculous amount of money because you didn't know about cost plans or didn't setup budget alerts.



Accidental AWS cost

There are currently more than 60 emulated AWS cloud services (and most of them free) provided by LocalStack. As an introduction we will create an S3 bucket and emulate the deployment of a static website and witness that it works just as if it were deployed on AWS S3.



AWS services offered by LocalStack

You can go to the [official LocalStack website](https://localstack.com) and log into your account or create a new one to see all the services provided by localStack on the dashboard. We can spin up and configure each and every service in the web dashboard but in our demo we will use the CLI to interact with our resources.

Installation

There are two ways to run LocalStack in your pc. You can install it using the python package manager (pip) on your machine or just use docker to run a LocalStack container.

1. Using PIP

```
python -m virtualenv venv  
  
venv/Scripts/activate
```

```
pip install localstack
```

2. Starting LocalStack with Docker

If you have docker desktop installed on your machine, make sure the docker daemon is running and run the command below.

```
docker run --rm -it -p 4566:4566 -p 4510-4559:4510-4559 localstack/localstack
```

For this article, I will use the first choice and install localstack using pip. We also need to install AWS Command Line Interface so that we can interact with the underlying emulated services.

To download the AWS Command Line Interface go to [AWS CLI Docs](#) and find AWS CLI MSI installer for Windows, then run the setup. After the installation is complete, then lets proceed and configure the CLI using the commands below.

```
aws configure
AWS Access Key ID:test
AWS Secret Access Key:test
Default region name :us-east-1
Default output format[None]
```

The next step is to run LocalStack. So open a terminal and run the following : -

```
localstack start -d
```



```
Windows PowerShell
(venv) PS D:\localstack> aws --endpoint-url=http://localhost:4566 s3api create-bucket --bucket testbucket
{
  "Location": "/testbucket"
}
(venv) PS D:\localstack>
```

Create S3 bucket named testbucket

Lets see our list of buckets.

```
aws --endpoint-url=http://localhost:4566 s3api list-buckets
```

```
Windows PowerShell
(venv) PS D:\localstack> aws --endpoint-url=http://localhost:4566 s3api list-buckets
{
  "Buckets": [
    {
      "Name": "testbucket",
      "CreationDate": "2023-08-17T14:25:43+00:00"
    }
  ],
  "Owner": {
    "DisplayName": "webfile",
    "ID": "75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a"
  }
}
(venv) PS D:\localstack>
```

List of available buckets

Create a test file and copy it to the created bucket.

```
aws --endpoint-url=http://localhost:4566 s3 cp test.txt s3://testbucket
```

```
Windows PowerShell
(venv) PS D:\localstack> aws --endpoint-url=http://localhost:4566 s3 cp test.txt s3://testbucket
upload: .\test.txt to s3://testbucket/test.txt
(venv) PS D:\localstack>
```

Copy a test.txt file to the bucket

Lets make sure the file is uploaded to the bucket.

```
aws --endpoint-url=http://localhost:4566 s3api list-objects --bucket testbucket
```

```
Windows PowerShell
(venv) PS D:\localstack> aws --endpoint-url=http://localhost:4566 s3api list-objects --bucket testbucket
{
  "Contents": [
    {
      "Key": "test.txt",
      "LastModified": "2023-08-18T12:32:27+00:00",
      "ETag": "\"f3b25701fe362ec84616a93a45ce9998\"",
      "Size": 2,
      "StorageClass": "STANDARD",
      "Owner": {
        "DisplayName": "webfile",
        "ID": "75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a"
      }
    }
  ]
}
```

list objects

Example 2 - Host a static website

In order to host a static website inside the S3 bucket we just created above, we need to prepare two html files: `index.html` and `error.html` and put them in a folder called `website`.

The `index.html` file serves as the main entry point for your website, representing the content and structure of your homepage.

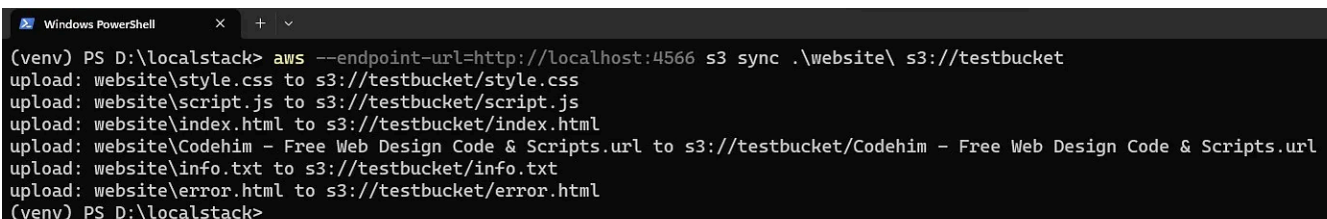
On the other hand, the `error.html` file is used to create custom error pages that gets displayed during HTTP errors like 404 (Not Found) or 403 (Forbidden).

We also can attach a bucket policy and allow public access to all of its contents. So create a file named `bucket_policy.json` in the root directory and add the following code:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::testbucket/*"
    }
  ]
}
```

Lets go ahead and sync the website folder that contains our files to the S3 bucket using the command below.

```
aws --endpoint-url=http://localhost:4566 s3 sync .\website\ s3://testbucket
```



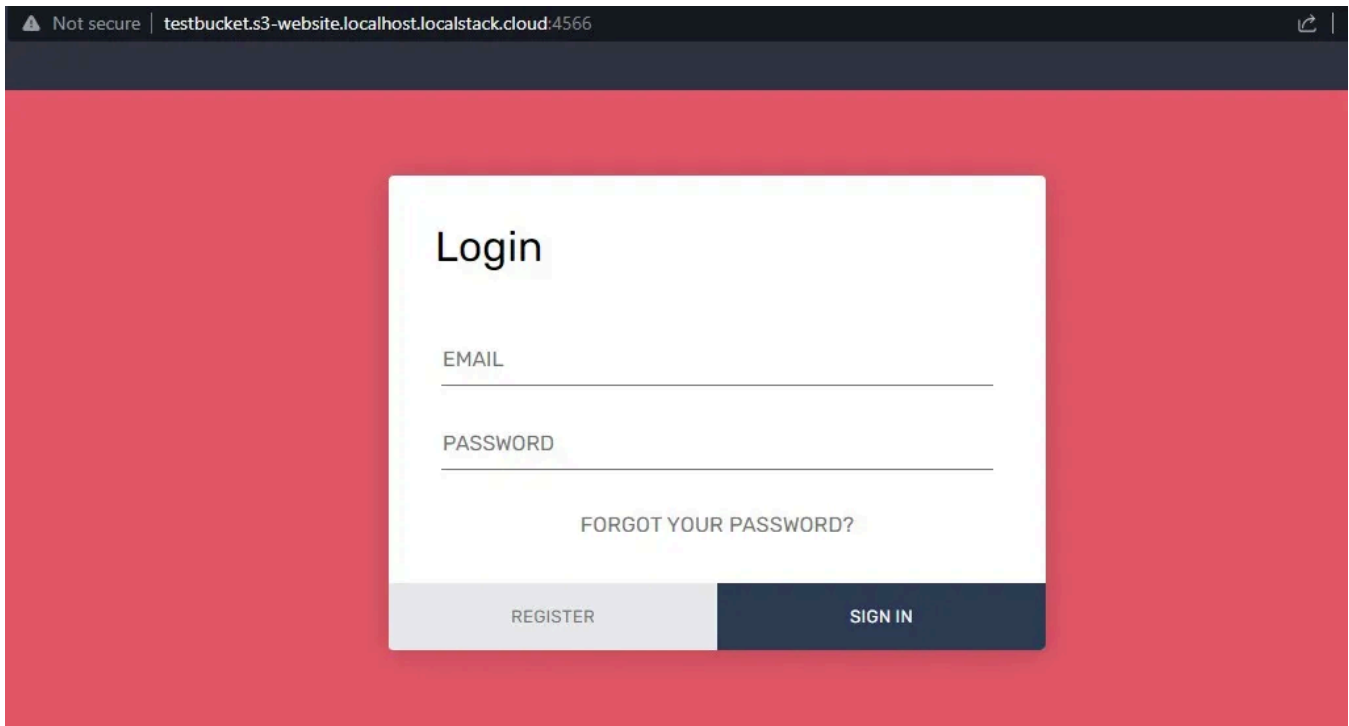
```
Windows PowerShell
(venv) PS D:\localstack> aws --endpoint-url=http://localhost:4566 s3 sync .\website\ s3://testbucket
upload: website\style.css to s3://testbucket/style.css
upload: website\script.js to s3://testbucket/script.js
upload: website\index.html to s3://testbucket/index.html
upload: website\Codehim - Free Web Design Code & Scripts.url to s3://testbucket/Codehim - Free Web Design Code & Scripts.url
upload: website\info.txt to s3://testbucket/info.txt
upload: website\error.html to s3://testbucket/error.html
(venv) PS D:\localstack>
```

Sync the website directory to the S3 bucket

Finally, we enable static website hosting on the bucket and configure the index and error documents:


```
aws --endpoint-url=http://localhost:4566 s3 website s3://testbucket/ --index-do
```

Our website is now hosted on the emulated S3 bucket. So let's browse to "http://testbucket.s3website.localstack.cloud:4566/" and see what it looks like.



index page

Something is amiss.

our error page

Conclusion

In this article we saw the capability of LocalStack in emulating a wide range of AWS services on how it can benefit to configure and test our infrastructure setups and also how it can be a tool to develop hands on AWS experience for beginners. If you have any doubts or questions regarding the article or want to know about other AWS services that can be emulated by LocalStack, feel free to share them in the comment section.