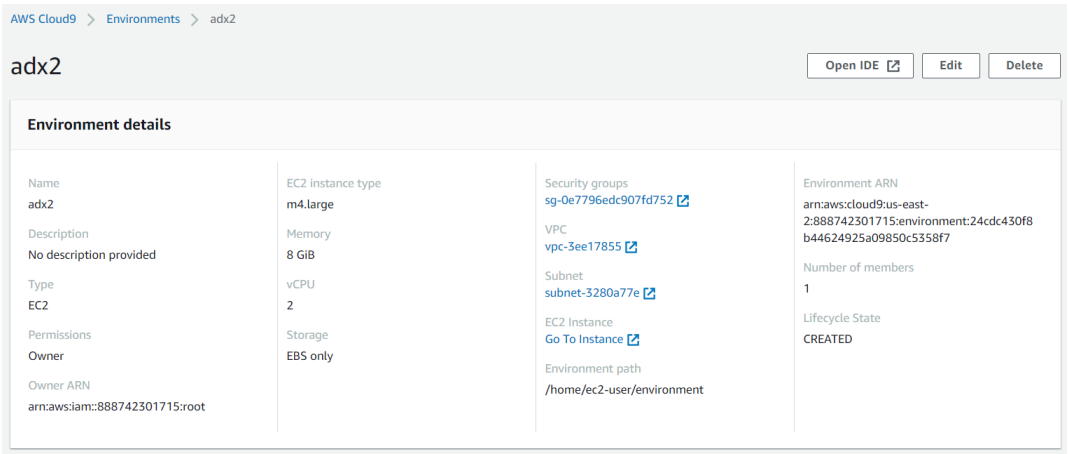
AWS How to host a static website on Amazon S3 using AWS SDK and AWS Cloud9

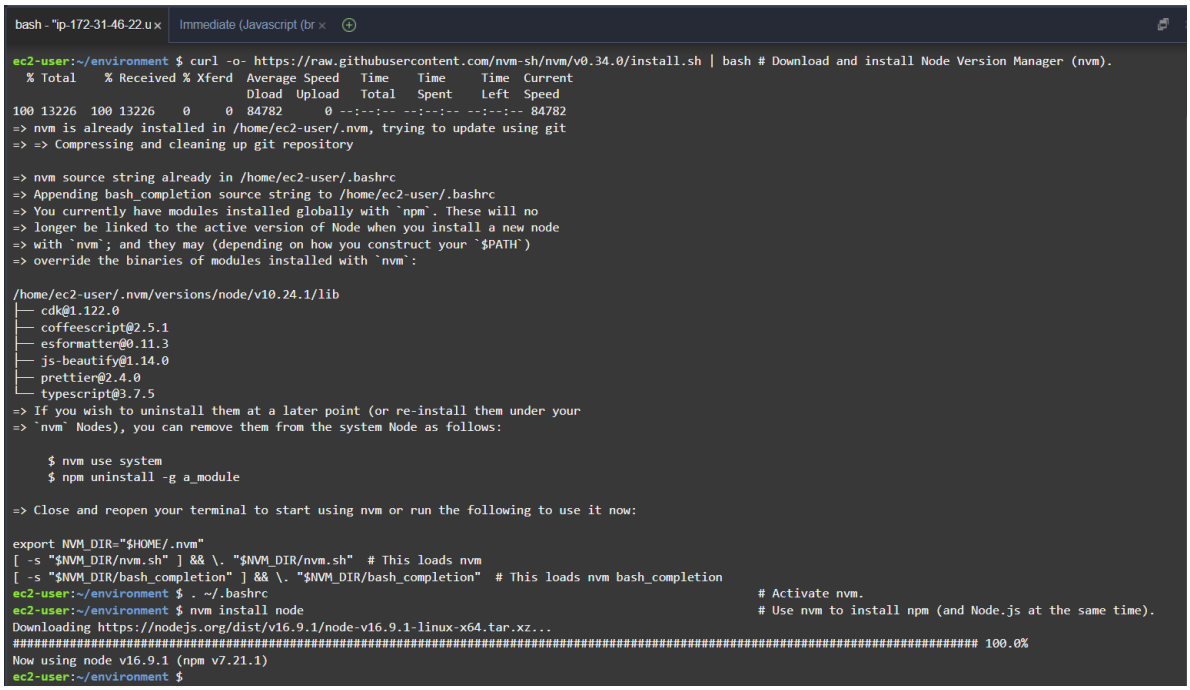
1. Environment Setup

- Set up AWS Cloud9 with the AWS SDK for JavaScript
- Step 1: Navigate to AWS Cloud9 and click 'Create environment'
    - Name: adx2
    - Environment type: Create a new EC2 instance for environment (direct access)
    - Instance type: m4.large (8GiB RAM + 2 vCPU)
    - Platform: Amazon Linux 2
    - Cost-saving setting: After 30 minutes (default)
    - Network (VPC): vpc-3ee17855 (default)
    - Subnet: subnet-3280a77e (default in us-east-2c)



- To set up the SDK for JavaScript, open Cloud9 IDE and set up Node.js as follows:

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash
. ~/.bashrc
nvm install node
```



- Install aws-sdk

```
npm install aws-sdk
```

```
ec2-user:~/environment $ npm install aws-sdk
npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy. new code should use the URLSearchParams API instead.
npm WARN deprecated uuid@3.3.2: Please upgrade  to version 7 or higher.  Older versions may use Math.random() in certain circumstances, whic
h is known to be problematic.  See https://v8.dev/blog/math-random for details.

added 14 packages, and audited 100 packages in 3s

3 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
ec2-user:~/environment $
```

- Install client-s3

```
npm install @aws-sdk/client-s3
```

```
ec2-user:~/environment $ npm install @aws-sdk/client-s3
npm WARN old lockfile
npm WARN old lockfile The package-lock.json file was created with an old version of npm,
npm WARN old lockfile so supplemental metadata must be fetched from the registry.
npm WARN old lockfile
npm WARN old lockfile This is a one-time fix-up, please be patient...
npm WARN old lockfile

added 23 packages, removed 4 packages, and audited 86 packages in 3s

2 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
ec2-user:~/environment $
```

- Install AWS CLI

```
pip3 install awscli --upgrade –user
```

```
ec2-user:~/environment $ pip3 install awscli --upgrade --user
Collecting awscli
  Downloading awscli-1.20.42-py3-none-any.whl (3.7 MB)
     |████████████████████████████████| 3.7 MB 4.9 MB/s
Collecting rsa<4.8,>=3.1.2
  Downloading rsa-4.7.2-py3-none-any.whl (34 kB)
Collecting PyYAML<5.5,>=3.10
  Downloading PyYAML-5.4.1-cp37-cp37m-manylinux1_x86_64.whl (636 kB)
     |████████████████████████████████| 636 kB 47.4 MB/s
Collecting botocore==1.21.42
  Downloading botocore-1.21.42-py3-none-any.whl (7.9 MB)
     |████████████████████████████████| 7.9 MB 105 kB/s
Collecting colorama<0.4.4,>=0.2.5
  Downloading colorama-0.4.3-py2.py3-none-any.whl (15 kB)
Requirement already satisfied, skipping upgrade: docutils<0.16,>=0.10 in /usr/lib/python3.7/site-packages (from awscli) (0.14)
Collecting s3transfer<0.6.0,>=0.5.0
  Downloading s3transfer-0.5.0-py3-none-any.whl (79 kB)
     |████████████████████████████████| 79 kB 15.4 MB/s
Collecting pyasn1>=0.1.3
  Downloading pyasn1-0.4.8-py2.py3-none-any.whl (77 kB)
     |████████████████████████████████| 77 kB 10.3 MB/s
Requirement already satisfied, skipping upgrade: jmespath<1.0.0,>=0.7.1 in /usr/local/lib/python3.7/site-packages (from botocore==1.21.42->awscli) (0.10.0)
Requirement already satisfied, skipping upgrade: urllib3<1.27,>=1.25.4 in /usr/local/lib/python3.7/site-packages (from botocore==1.21.42->awscli) (1.26.6)
Requirement already satisfied, skipping upgrade: python-dateutil<3.0.0,>=2.1 in /usr/local/lib/python3.7/site-packages (from botocore==1.21.42->awscli) (2.8.2)
Requirement already satisfied, skipping upgrade: six>=1.5 in /usr/local/lib/python3.7/site-packages (from python-dateutil<3.0.0,>=2.1->botocore==1.21.42->awscli) (1
.16.0)
Installing collected packages: pyasn1, rsa, PyYAML, botocore, colorama, s3transfer, awscli
Successfully installed PyYAML-5.4.1 awscli-1.20.42 botocore-1.21.42 colorama-0.4.3 pyasn1-0.4.8 rsa-4.7.2 s3transfer-0.5.0
ec2-user:~/environment $
```

- Add the AWS CLI Version 1 executable to the command line path

```
export PATH=~/usr/bin:$PATH
source ~/.bash_profile
```

```
bash - "ip-172-31-46-22.u ×   Immediate (Javascript (br ×
ec2-user:~/environment $ which aws
/usr/bin/aws
ec2-user:~/environment $ which python
alias python='python3'
        /usr/bin/python3
ec2-user:~/environment $ echo $SHELL
/bin/bash
ec2-user:~/environment $ export PATH=~/usr/bin:$PATH
ec2-user:~/environment $ source ~/.bash_profile
ec2-user:~/environment $
```

- Download and unzip the file Static_Website.zip from GitHub

```
wget https://github.com/nooruzaman/CSE2ADX_A2/raw/main/Static_Website.zip
unzip Static_Website
```

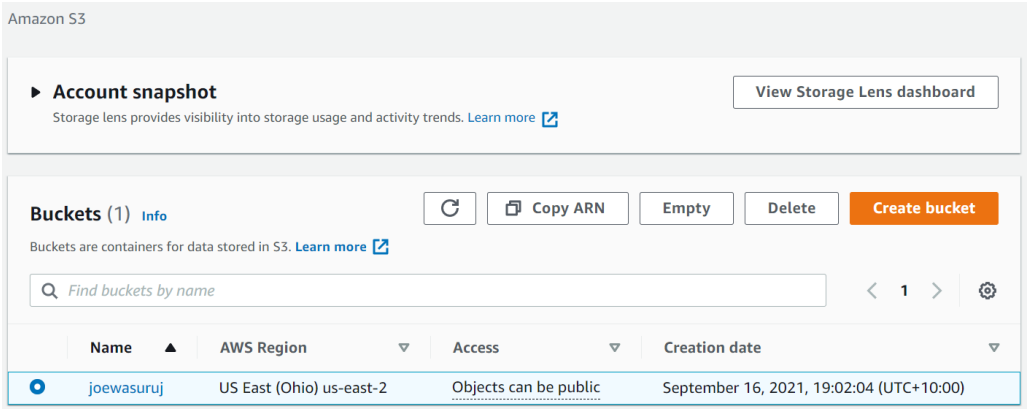
```
ec2-user:~/environment $ wget https://github.com/nooruzaman/CSE2ADX_A2/raw/main/Static_Website.zip
--2021-09-16 10:37:30--  https://github.com/nooruzaman/CSE2ADX_A2/raw/main/Static_Website.zip
Resolving github.com (github.com)... 140.82.113.4
Connecting to github.com (github.com)|140.82.113.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/nooruzaman/CSE2ADX_A2/main/Static_Website.zip [following]
--2021-09-16 10:37:30--  https://raw.githubusercontent.com/nooruzaman/CSE2ADX_A2/main/Static_Website.zip
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 412058 (402K) [application/zip]
Saving to: ‘Static_Website.zip’

100%[=====================================================================================>] 412,058     --.-K/s   in 0.04s

2021-09-16 10:37:31 (11.2 MB/s) - ‘Static_Website.zip’ saved [412058/412058]

ec2-user:~/environment $ unzip Static_Website
Archive:  Static_Website.zip
  inflating: Static_Website/index.html
  inflating: Static_Website/solar-system.jpg
  inflating: Static_Website/space2.jpg
ec2-user:~/environment $
ec2-user:~/environment $ ls
aws-doc-sdk-examples  README.md  Static_Website  Static_Website.zip
ec2-user:~/environment $
```

2. Bucket Creation

- Create an Amazon S3 bucket using the following commands:

```
touch s3createbucket.js
cat << EOF > s3createbucket.js
// Load the AWS SDK for Node.js
var AWS = require('aws-sdk');
// Set the region
AWS.config.update({region: 'us-east-2'});
// Create S3 service object
s3 = new AWS.S3({apiVersion: '2006-03-01'});
// Create the parameters for calling createBucket
var bucketParams = {
  Bucket : process.argv[2]
};
// call S3 to create the bucket
s3.createBucket(bucketParams, function(err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data.Location);
  }
});
EOF
```

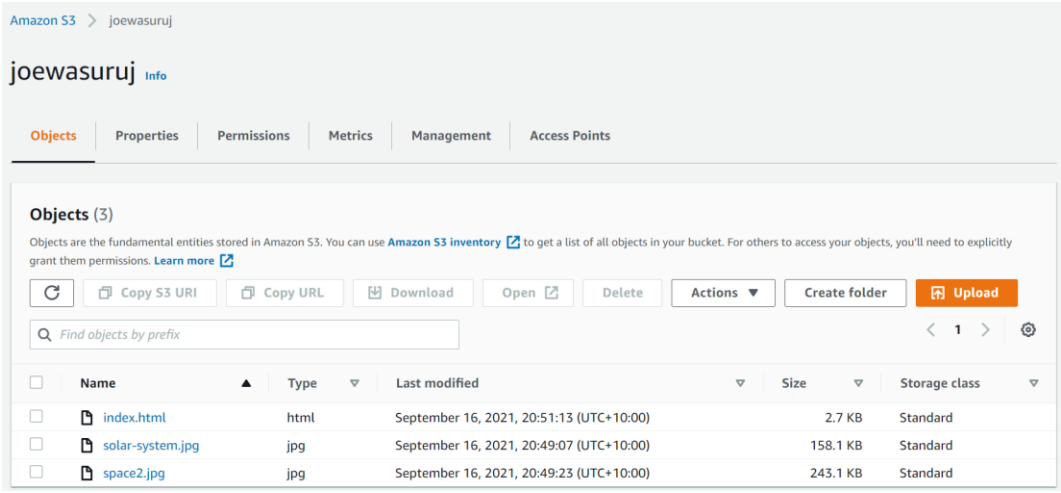
```
node s3createbucket.js joewasuruj
```



- Upload the code files from static-website.zip

```
touch s3upload.js
cat << EOF > s3upload.js
// Load the AWS SDK for Node.js
var AWS = require('aws-sdk');
// Set the region
AWS.config.update({region: 'us-east-2'});
// Create S3 service object
var s3 = new AWS.S3({apiVersion: '2006-03-01'});
// call S3 to retrieve upload file to specified bucket
var uploadParams = {Bucket: process.argv[2], Key: '', Body: ''};
var file = process.argv[3];
// Configure the file stream and obtain the upload parameters
var fs = require('fs');
var fileStream = fs.createReadStream(file);
fileStream.on('error', function(err) {
  console.log('File Error', err);
});
uploadParams.Body = fileStream;
var path = require('path');
uploadParams.Key = path.basename(file);
// call S3 to retrieve upload file to specified bucket
s3.upload (uploadParams, function (err, data) {
  if (err) {
    console.log("Error", err);
  } if (data) {
    console.log("Upload Success", data.Location);
  }
});
EOF
```

```
node s3upload.js joewasuruj ~/environment/Static_Website/index.html
node s3upload.js joewasuruj ~/environment/Static_Website/solar-system.jpg
node s3upload.js joewasuruj ~/environment/Static_Website/space2.jpg
```



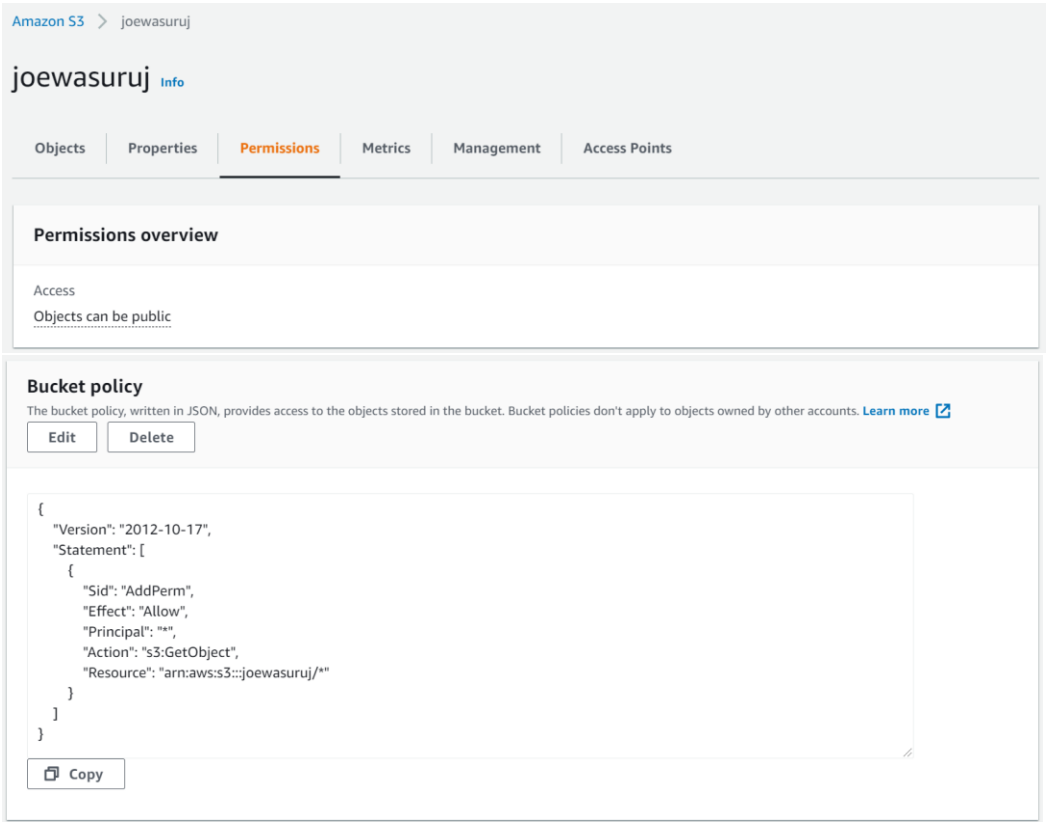
3. Creation of Bucket Policy

- Create an S3 bucket policy to allow everyone to access S3 bucket
- Apply to code to my bucket joewasuruj

```
touch s3setbucketpolicy.js
cat << EOF > s3setbucketpolicy.js
// Load the AWS SDK for Node.js
var AWS = require('aws-sdk');
// Set the region
AWS.config.update({region: 'us-east-2'});
// Create S3 service object
s3 = new AWS.S3({apiVersion: '2006-03-01'});
var readOnlyAnonUserPolicy = {
 Version: "2012-10-17",
 Statement: [
  {
   Sid: "PublicReadGetObject",
   Effect: "Allow",
   Principal: "*",
   Action: [
    "s3:GetObject"
   ],
   Resource: [
    ""
   ]
  }
 ]
};
// create selected bucket resource string for bucket policy
var bucketResource = "arn:aws:s3:::" + process.argv[2] + "/*";
readOnlyAnonUserPolicy.Statement[0].Resource[0] = bucketResource;
// convert policy JSON into string and assign into params
var bucketPolicyParams = {Bucket: process.argv[2], Policy: JSON.stringify(readOnlyAnonUserPolicy)};
// set the new policy on the selected bucket
```

```
s3.putBucketPolicy(bucketPolicyParams, function(err, data) {
  if (err) {
    // display error message
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
EOF
```

```
node s3setbucketpolicy.js joewasuruj
```



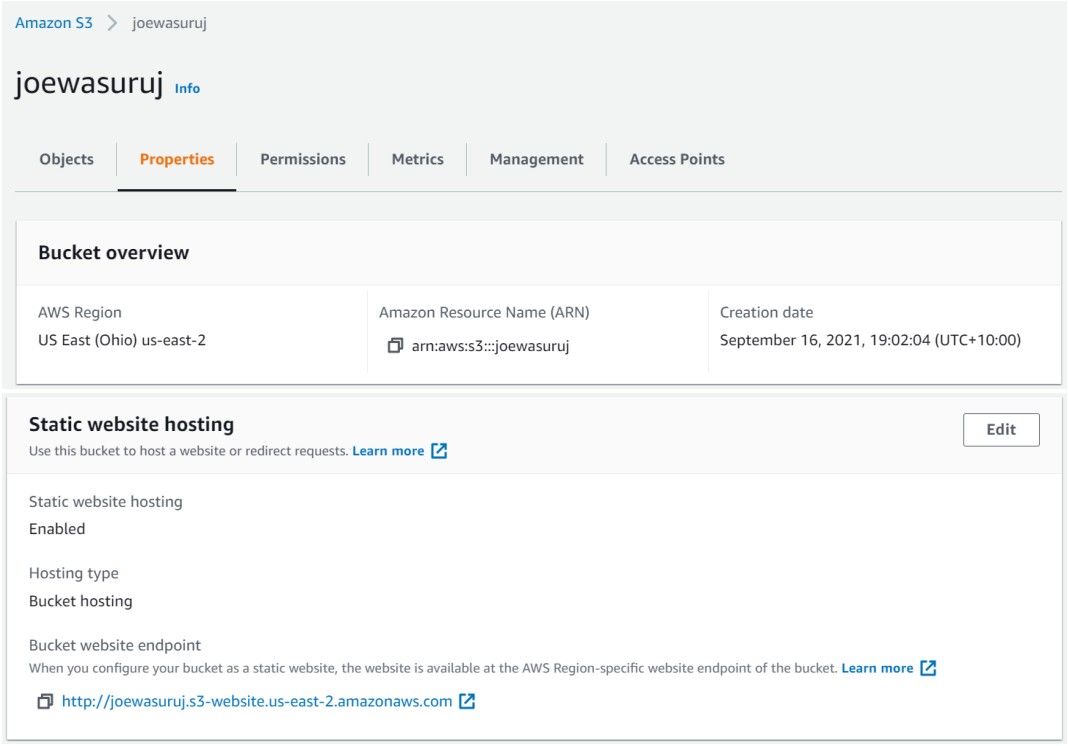
4. Enable Web Hosting

- Enable webhosting on the joewasuruj bucket

```
touch s3setbucketwebsite.js
cat << EOF > s3setbucketwebsite.js
// Load the AWS SDK for Node.js
var AWS = require('aws-sdk');
// Set the region
AWS.config.update({region: 'us-east-2'});
// Create S3 service object
s3 = new AWS.S3({apiVersion: '2006-03-01'});
// Create JSON for putBucketWebsite parameters
var staticHostParams = {
  Bucket: '',
```

```
  WebsiteConfiguration: {
    ErrorDocument: {
      Key: ''
    },
    IndexDocument: {
      Suffix: ''
    },
  }
};
// Insert specified bucket name and index and error documents into params JSON
// from command line arguments
staticHostParams.Bucket = process.argv[2];
staticHostParams.WebsiteConfiguration.IndexDocument.Suffix = process.argv[3];
staticHostParams.WebsiteConfiguration.ErrorDocument.Key = process.argv[4];
// set the new website configuration on the selected bucket
s3.putBucketWebsite(staticHostParams, function(err, data) {
  if (err) {
    // display error message
    console.log("Error", err);
  } else {
    // update the displayed website configuration for the selected bucket
    console.log("Success", data);
  }
});
EOF
```

```
node s3setbucketwebsite.js joewasuruj index.html index.html
```



https://github.com/joe-wasuruj/aws-step-by-step-guides

5. Testing of Website

- Before testing the website, ensure that “index.html” has the correct content type
    - In the joewasuruj bucket, click “index.html”
    - Click “Properties”, scroll down to “Metadata”, click “Edit”
    - Ensure the “Content-Type” is “text/html”
- Test the website using the bucket endpoint:
    - http://joewasuruj.s3-website.us-east-2.amazonaws.com/