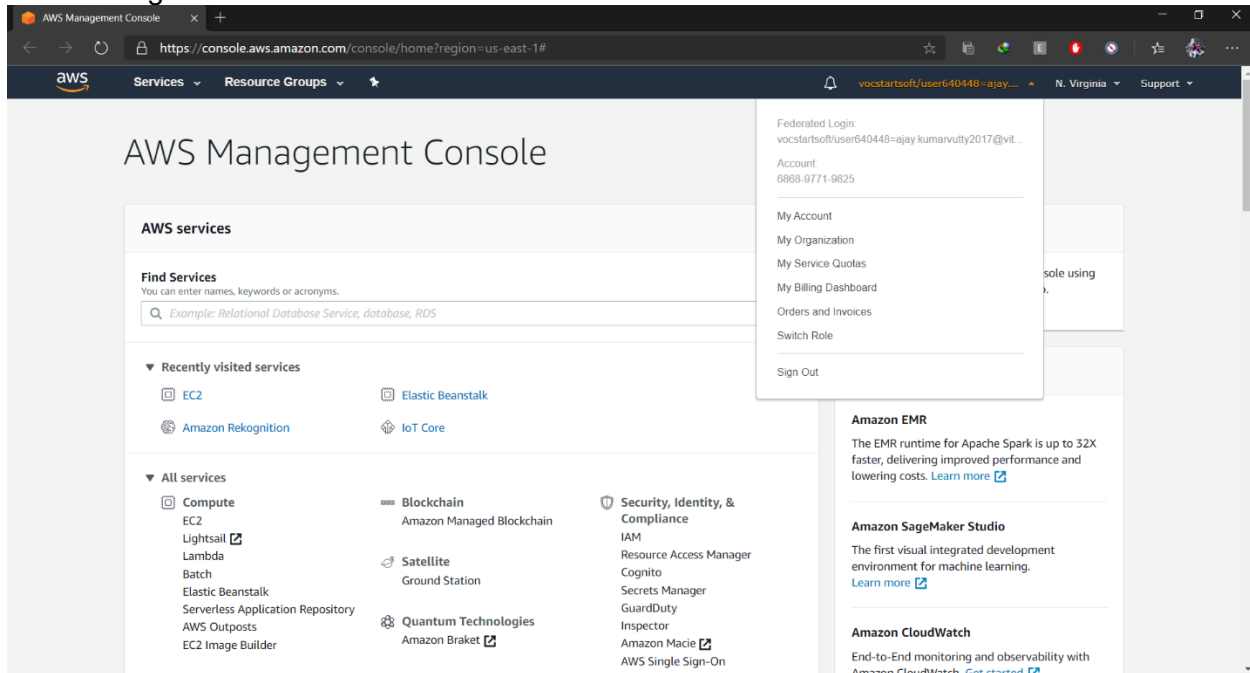
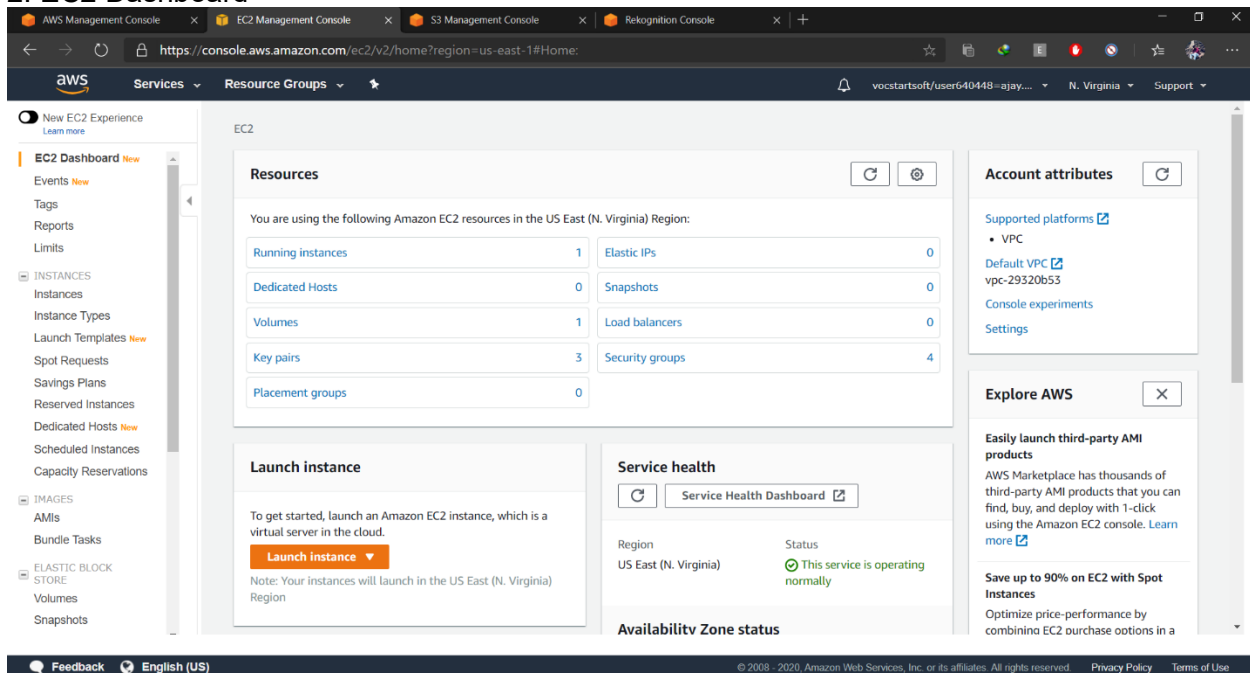


Screenshots needed for Dashboards

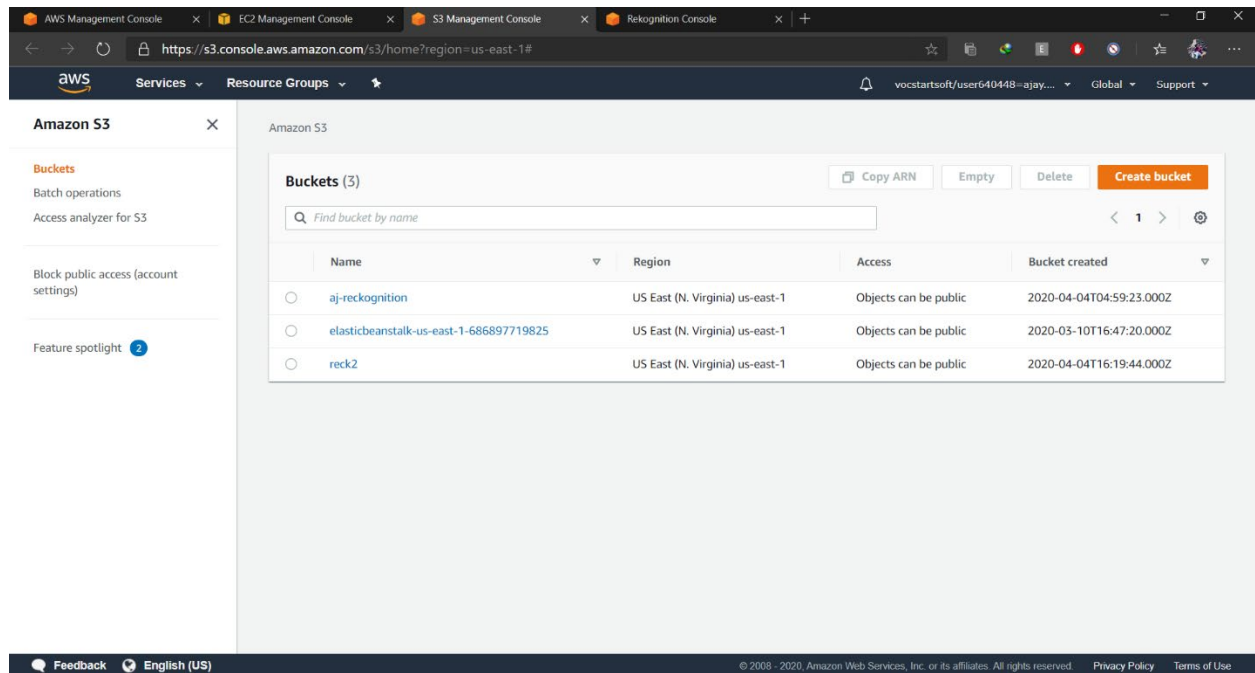
1. AWS Login screen with username



2. EC2 Dashboard



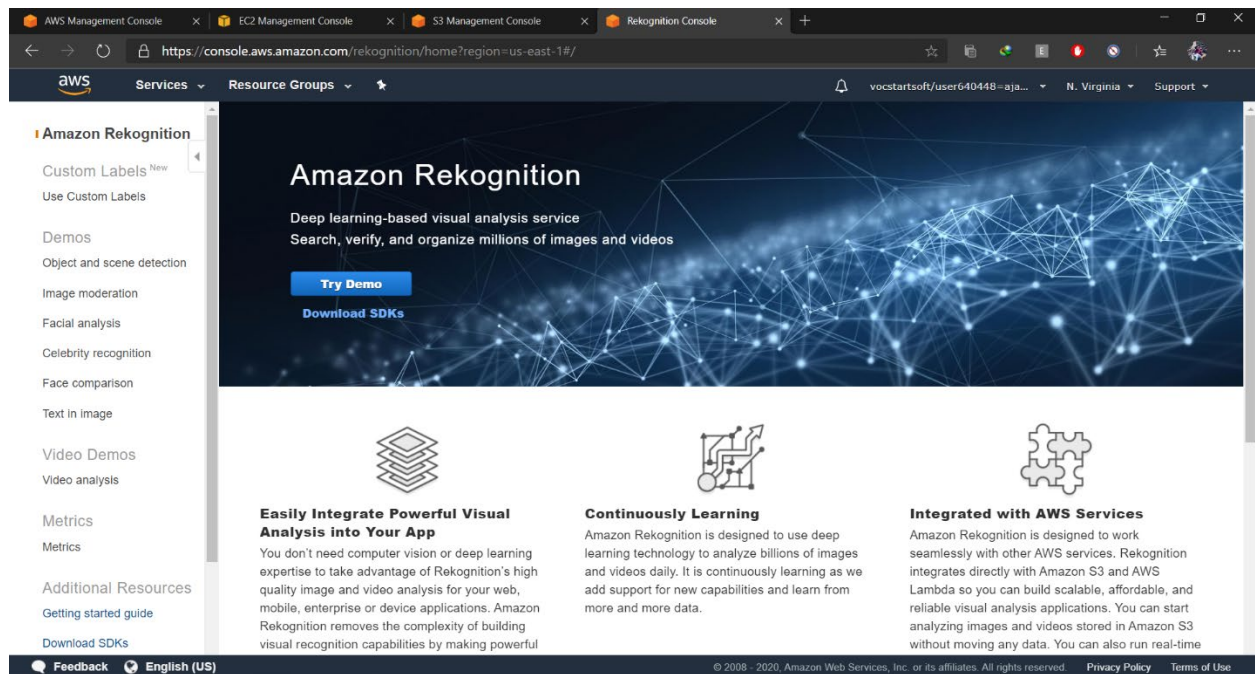
3. S3 Dashboard



The screenshot shows the Amazon S3 console interface. The left sidebar contains navigation links for Buckets, Batch operations, Access analyzer for S3, Block public access (account settings), and Feature spotlight. The main content area displays a table of buckets with columns for Name, Region, Access, and Bucket created. The table lists three buckets: 'aj-rekognition', 'elasticbeanstalk-us-east-1-686897719825', and 'reck2', all in the US East (N. Virginia) region with public access.

Name	Region	Access	Bucket created
aj-rekognition	US East (N. Virginia) us-east-1	Objects can be public	2020-04-04T04:59:23.000Z
elasticbeanstalk-us-east-1-686897719825	US East (N. Virginia) us-east-1	Objects can be public	2020-03-10T16:47:20.000Z
reck2	US East (N. Virginia) us-east-1	Objects can be public	2020-04-04T16:19:44.000Z

4. Rekognition Dashboard



The screenshot shows the Amazon Rekognition console home page. The left sidebar contains navigation links for Custom Labels, Demos, Object and scene detection, Image moderation, Facial analysis, Celebrity recognition, Face comparison, Text in image, Video Demos, Video analysis, Metrics, and Additional Resources. The main content area features a large header with the text 'Amazon Rekognition' and 'Deep learning-based visual analysis service'. Below the header are three sections: 'Easily Integrate Powerful Visual Analysis into Your App', 'Continuously Learning', and 'Integrated with AWS Services'.

Amazon Rekognition
Deep learning-based visual analysis service
Search, verify, and organize millions of images and videos

[Try Demo](#)
[Download SDKs](#)

Easily Integrate Powerful Visual Analysis into Your App
You don't need computer vision or deep learning expertise to take advantage of Rekognition's high quality image and video analysis for your web, mobile, enterprise or device applications. Amazon Rekognition removes the complexity of building visual recognition capabilities by making powerful

Continuously Learning
Amazon Rekognition is designed to use deep learning technology to analyze billions of images and videos daily. It is continuously learning as we add support for new capabilities and learn from more and more data.

Integrated with AWS Services
Amazon Rekognition is designed to work seamlessly with other AWS services. Rekognition integrates directly with Amazon S3 and AWS Lambda so you can build scalable, affordable, and reliable visual analysis applications. You can start analyzing images and videos stored in Amazon S3 without moving any data. You can also run real-time

Screenshots needed for EC2

1. Choosing an AMI

Launch instance wizard | EC2 M... x

https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard:

Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

Quick Start

My AMIs

AWS Marketplace

Community AMIs

Free tier only

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0fc61db8544a617ed (64-bit x86) / ami-0f90a34c9df977efb (64-bit Arm)

Amazon Linux 2.29.1, and the latest software packages through extras.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

64-bit (x86)

64-bit (Arm)

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-09a5b0b7edf08843d

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

64-bit (x86)

Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-0c322300a1dd5dc79 (64-bit x86) / ami-03587fa4048e9eb92 (64-bit Arm)

Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

64-bit (x86)

64-bit (Arm)

Feedback English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

2. Choosing an Instance Type

Launch instance wizard | EC2 M... x

https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard:

Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All Instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

Feedback English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

3. Adding Storage

Launch instance wizard | EC2 M...

https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard

Services Resource Groups

vocstartsoft/user640448--ajay... N. Virginia Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type ⓘ	Device ⓘ	Snapshot ⓘ	Size (GiB) ⓘ	Volume Type ⓘ	IOPS ⓘ	Throughput (MB/s) ⓘ	Delete on Termination ⓘ	Encryption ⓘ
Root	/dev/xvda	snap-0e27a39c6e2f9f079	8	General Purpose SSD (gp2) ▾	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted ▾

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Cancel Previous Review and Launch Next: Add Tags

Feedback English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

4. Configuring Security Group

Launch instance wizard | EC2 M...

https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard

Services Resource Groups

vocstartsoft/user640448--ajay... N. Virginia Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

Description:

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ
SSH ▾	TCP	22	Custom ▾ 0.0.0.0/0	e.g. SSH for Admin Desktop

Add Rule

Warning

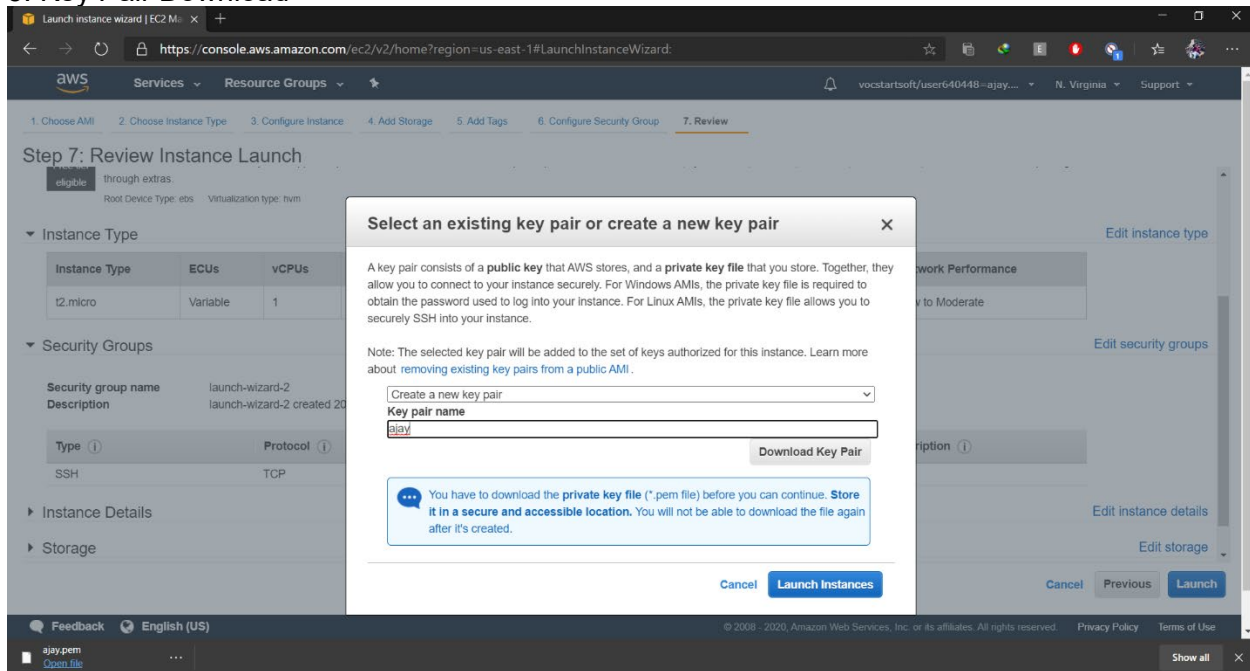
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Previous Review and Launch

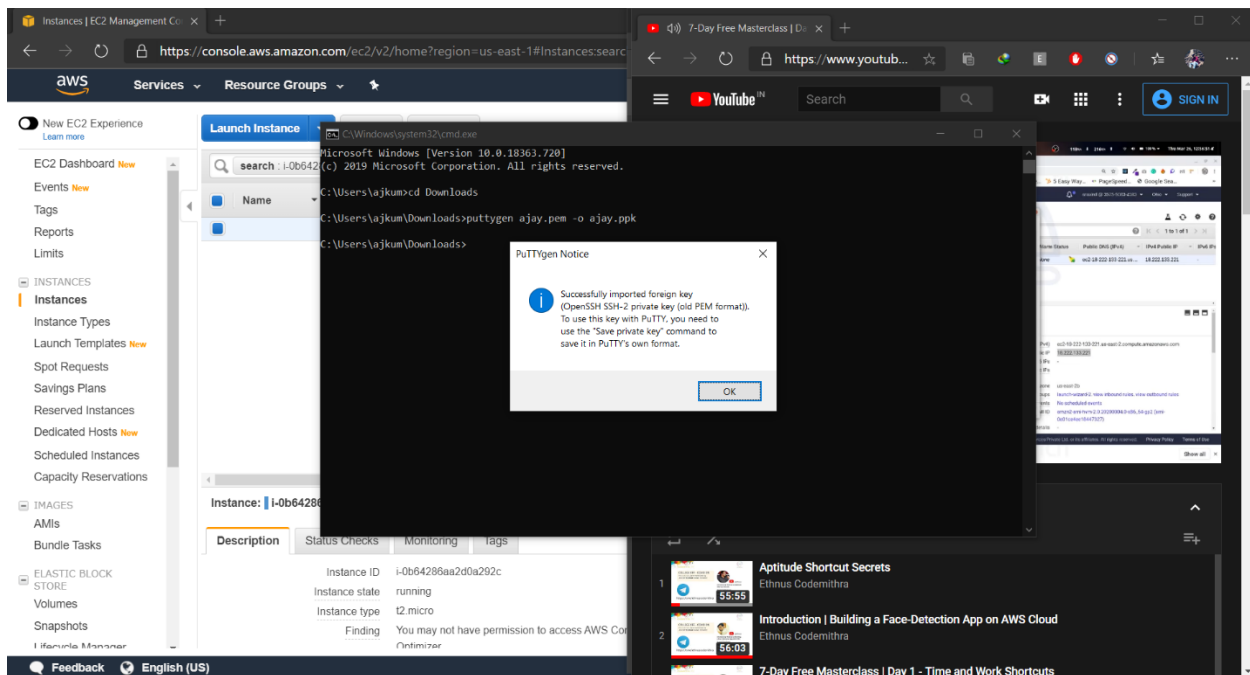
Feedback English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

5. Key Pair Download



6. PuTTYgen conversion from pem to ppk



7. Logged in EC2 black screen

The screenshot shows the AWS Management Console for an EC2 instance. A terminal window is open, displaying the login process for the 'ec2-user' on an Amazon Linux 2 AMI. The terminal output shows the user is logged in and the system is running. The instance details page is visible in the background, showing the instance ID 'i-0b64286aa2d0a292c' and its public DNS address 'ec2-18-206-120-91.compute-1.amazonaws.com'.

```
ec2-user@ip-172-31-95-125:~$ login as: ec2-user
ec2-user@ip-172-31-95-125:~$ ssh-keygen -f /dev/null -C ec2-user -P '' -m PEM
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ec2-user/.ssh/id_rsa
Your public key has been saved in /home/ec2-user/.ssh/id_rsa.pub
The key fingerprint is:
SHA256: 12:34:56:78:90:ab:cd:ef:12:34:56:78:90:ab:cd:ef
The key's randomart image is:
[+]

```

Instance: i-0b64286aa2d0a292c Public DNS: ec2-18-206-120-91.compute-1.amazonaws.com

Description	Status Checks	Monitoring	Tags
Instance ID	i-0b64286aa2d0a292c	Public DNS (IPv4)	ec2-18-206-120-91.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	18.206.120.91
Instance type	t2.micro	IPv6 IPs	-
Finding	You may not have permission to access AWS Compute Optimizer	Elastic IPs	-

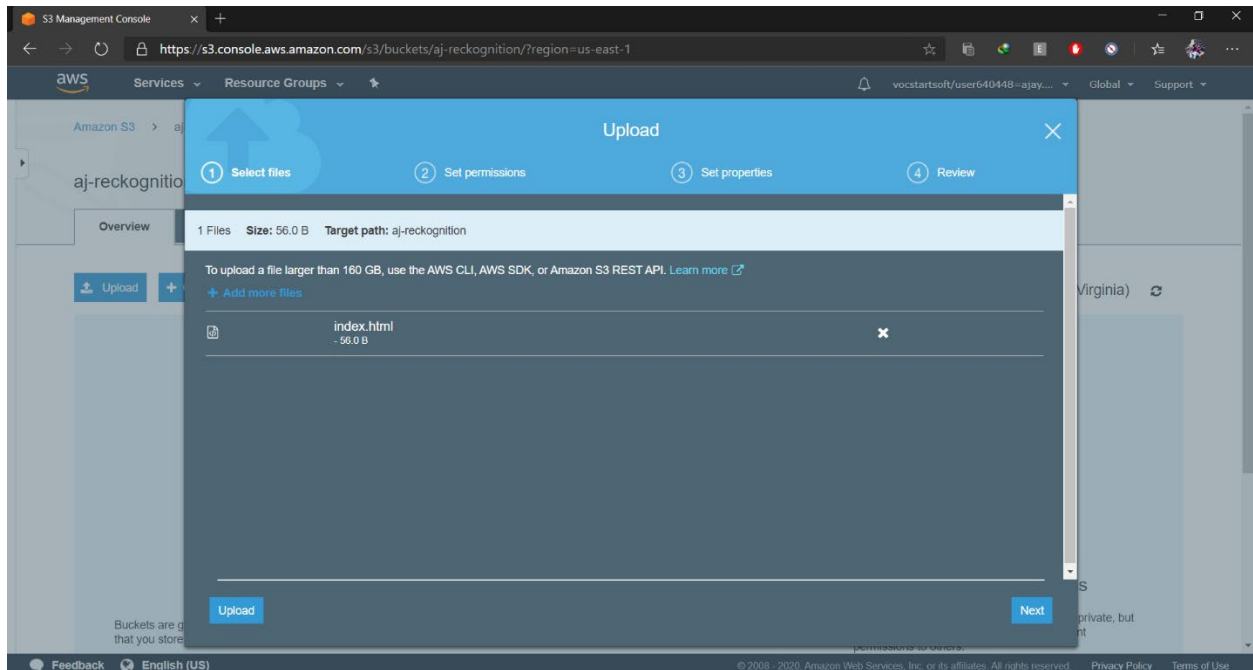
Screenshots needed for S3

1. Creating a bucket

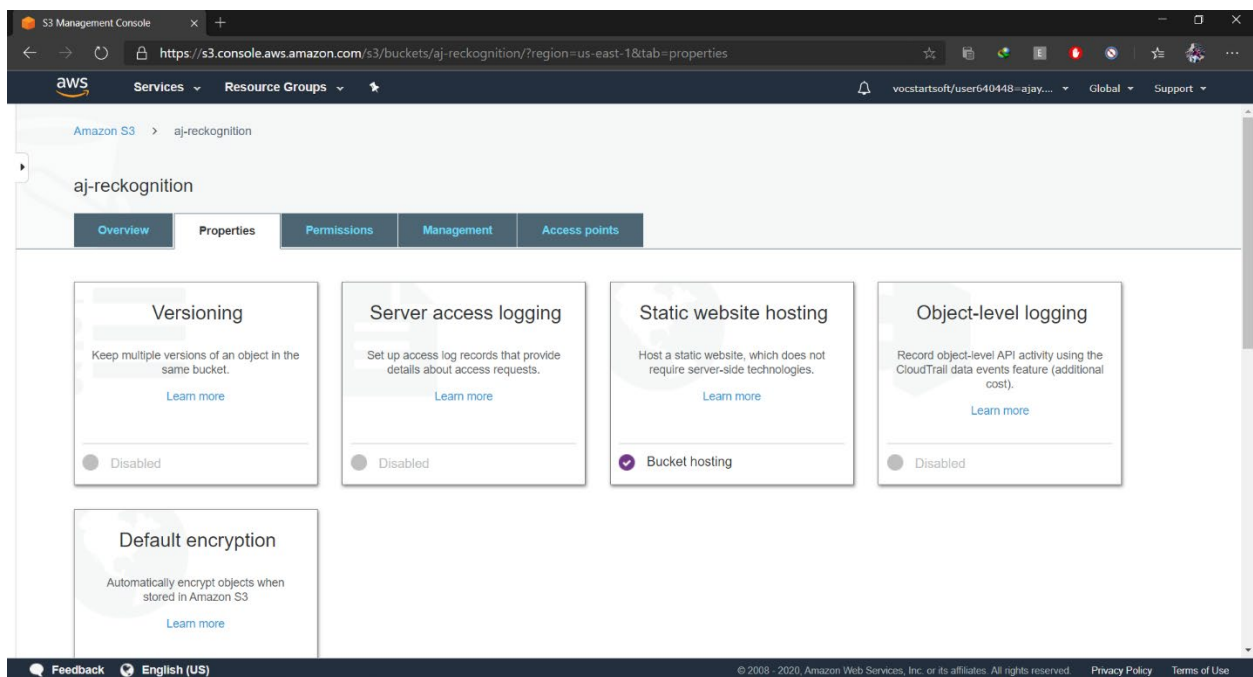
The screenshot shows the AWS S3 Management Console. The 'Buckets' section is active, displaying a list of existing buckets. The table shows two buckets: 'aj-reckognition' and 'elasticbeanstalk-us-east-1-686897719825'. The 'Create bucket' button is visible in the top right corner.

Name	Region	Access	Bucket created
aj-reckognition	US East (N. Virginia) us-east-1	Not Public	2020-04-04T04:59:23.000Z
elasticbeanstalk-us-east-1-686897719825	US East (N. Virginia) us-east-1	Objects can be public	2020-03-10T16:47:20.000Z

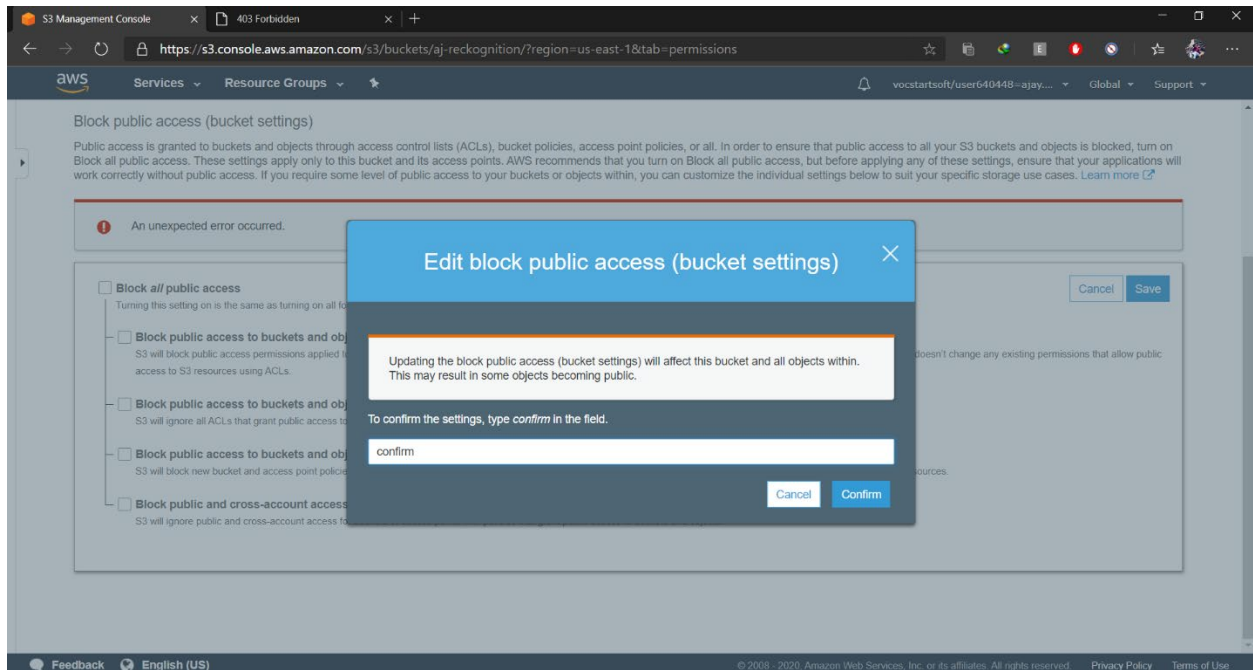
2. Uploading an Object



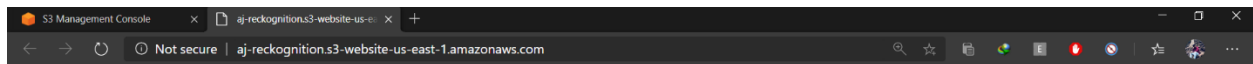
3. Enabling Static Website



4. Making the Object Public



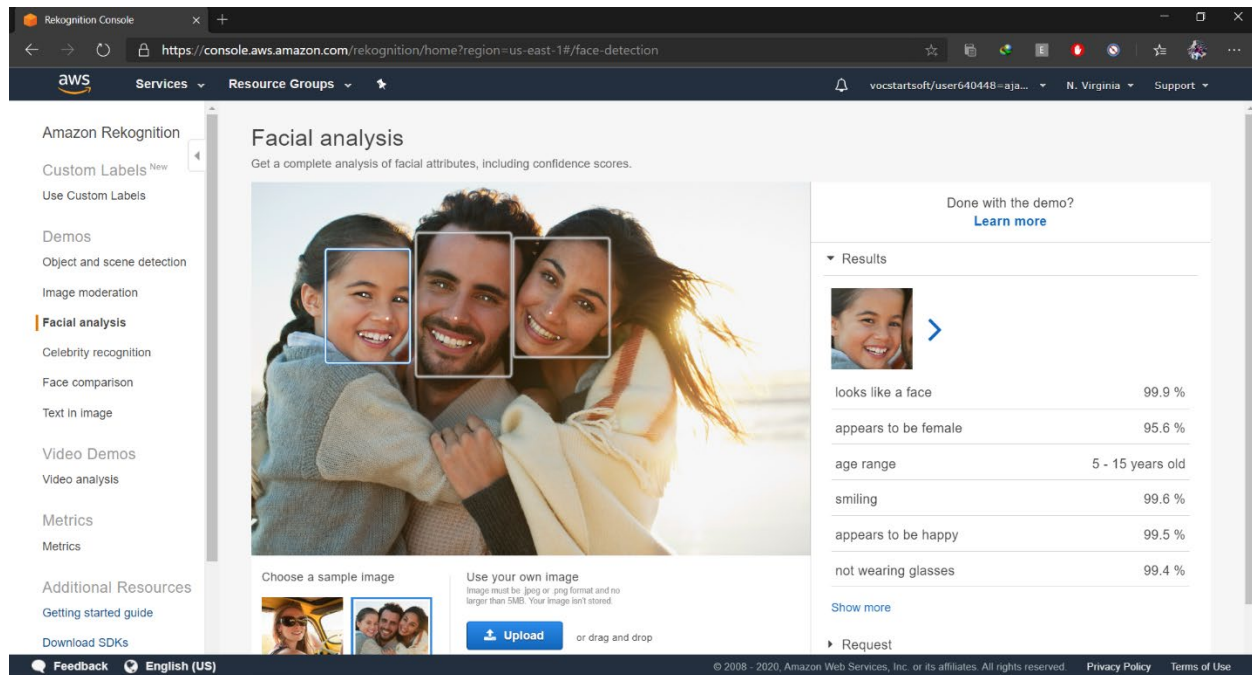
5. Checking the S3 link on the browser



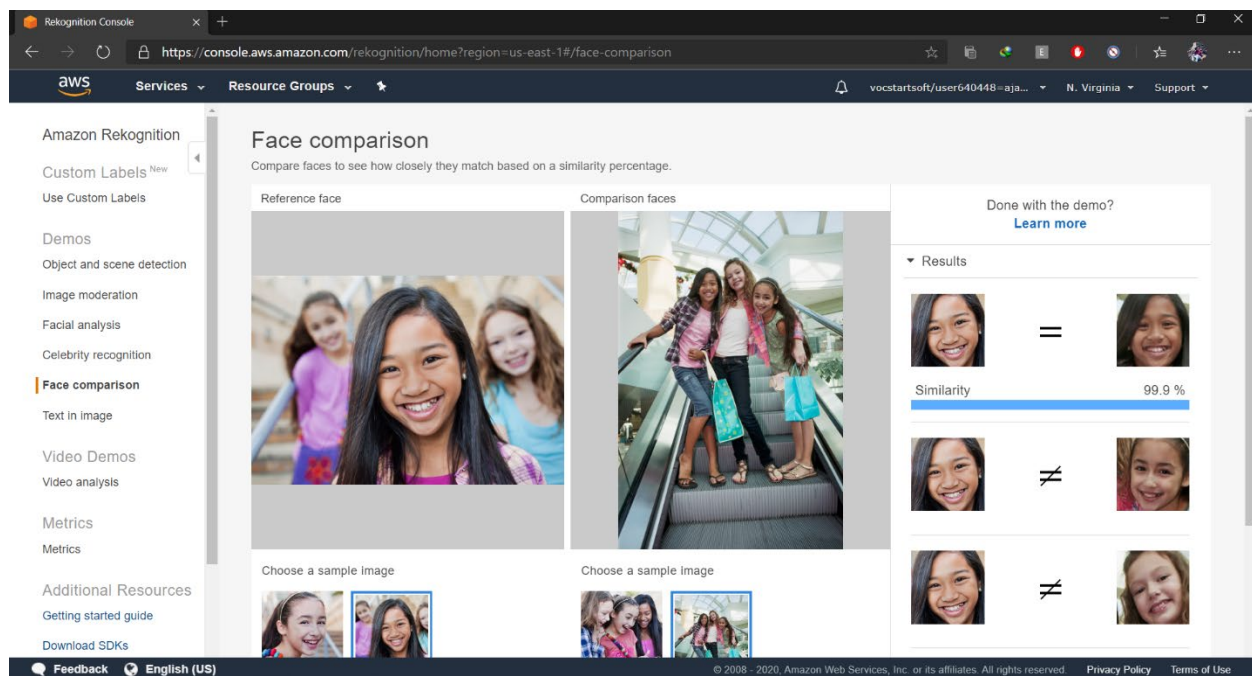
Hello AJAY!!!

Screenshots needed for Rekognition

1. Face Detect



2. Face Compare



3. Celebrity Recognition

The screenshot shows the Amazon Rekognition Console interface for the Celebrity Recognition demo. The left sidebar lists various services, with 'Celebrity recognition' highlighted. The main content area displays a large image of Jeff Bezos with a blue bounding box around his face. Below the image, there are options to 'Choose a sample image' or 'Use your own image'. The right sidebar shows the results of the recognition, including a small image of Jeff Bezos, his name, and a 'Match confidence' of 100%.

Amazon Rekognition

Custom Labels ^{New}

Use Custom Labels

Demos

Object and scene detection

Image moderation

Facial analysis

Celebrity recognition

Face comparison

Text in image

Video Demos

Video analysis

Metrics

Metrics

Additional Resources

Getting started guide

Download SDKs

Feedback English (US)


© 2008 - 2020 Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Celebrity recognition

Rekognition automatically recognizes celebrities in images and provides confidence scores.

Done with the demo? [Learn more](#)

▼ Results

 **Jeff Bezos** [Learn More](#)

Match confidence 100 %

► Request

► Response

Choose a sample image

Use your own image
Image must be .jpeg or .png format and no larger than 5MB. Your image isn't stored.

[Upload](#) or drag and drop

4. Text in Image

The screenshot shows the Amazon Rekognition Console interface for the Text in Image demo. The left sidebar lists various services, with 'Text in image' highlighted. The main content area displays a large image of a coffee cup with a smiley face and the text 'IT'S MONDAY but keep Smiling'. Below the image, there are options to 'Choose a sample image' or 'Use your own image'. The right sidebar shows the results of the text detection, including a list of detected text elements and their bounding boxes.

Amazon Rekognition

Custom Labels ^{New}

Use Custom Labels

Demos

Object and scene detection

Image moderation

Facial analysis

Celebrity recognition

Face comparison

Text in image

Video Demos

Video analysis

Metrics

Metrics

Additional Resources

Getting started guide

Download SDKs

Feedback English (US)

© 2008 - 2020 Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Text in image

Rekognition automatically detects and extracts text in your images. [Learn More](#)

Done with the demo? [Learn more](#)

▼ Results US English only

| IT'S |
| MONDAY |
| but keep |
| Smiling |

► Request

► Response

Choose a sample image

Use your own image
Image must be .jpeg or .png format and no larger than 5MB. Your image isn't stored.

[Upload](#) or drag and drop

Screenshots needed for EC2 & S3

1. Installing aws-sdk

```
ec2-user@ip-172-31-85-248:/var/www/html/face
require [--dev] [-r profile-source] [-p prefer-dist] [-f fixed] [--no-progress] [--no-suggests] [--no-update] [--no-scripts] [-u update-no-dev] [-u update-with-dependencies] [-u update-with-all-dependencies] [--ignore-platform-reqs] [--prefer-stable] [--prefer-lowest] [--sort-packages] [-o] [--optimize-autoloader] [-a|--classmap-authoritative] [--apcu-autoloader] [--] [<packages>]...

[ec2-user@ip-172-31-85-248 face]$
[ec2-user@ip-172-31-85-248 face]$ sudo /bin/dd if=/dev/zero of=/var/swap.1 bs=1M count=1024

1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB) copied, 13.4599 s, 79.8 MB/s
[ec2-user@ip-172-31-85-248 face]$
[ec2-user@ip-172-31-85-248 face]$
[ec2-user@ip-172-31-85-248 face]$
[ec2-user@ip-172-31-85-248 face]$ sudo /sbin/mkswap /var/swap.1
mkswap: /var/swap.1: insecure permissions 0644, 0600 suggested.
Setting up swapspace version 1, size = 1024 MiB (1073737728 bytes)
no label, UUID=71405310-b7b7-4ffe-89d6-bb485e9a7691
[ec2-user@ip-172-31-85-248 face]$ sudo /sbin/mkswap /var/swap.1
mkswap: /var/swap.1: insecure permissions 0644, 0600 suggested.
mkswap: /var/swap.1: warning: wiping old swap signature.
Setting up swapspace version 1, size = 1024 MiB (1073737728 bytes)
no label, UUID=71405310-b7b7-4ffe-89d6-bb485e9a7691
[ec2-user@ip-172-31-85-248 face]$ sudo /sbin/swapon /var/swap.1
swapon: /var/swap.1: insecure permissions 0644, 0600 suggested.
[ec2-user@ip-172-31-85-248 face]$ sudo php -d memory_limit=-1 ~/composer.phar require aws/aws-sdk-php
Using version ^2.8 for aws/aws-sdk-php
./composer.json has been created
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 3 installs, 0 updates, 0 removals
  - Installing symfony/event-dispatcher (v2.8.52): Loading from cache
  - Installing guzzle/guzzle (v3.9.3): Downloading (100%)
  - Installing aws/aws-sdk-php (2.8.31): Downloading (100%)
symfony/event-dispatcher suggests installing symfony/dependency-injection
symfony/event-dispatcher suggests installing symfony/http-kernel
guzzle/guzzle suggests installing guzzlehttp/guzzle (Guzzle 5 has moved to a new package name
. The package you have installed, Guzzle 3, is deprecated.)
aws/aws-sdk-php suggests installing doctrine/cache (Adds support for caching of credentials a
nd responses)
aws/aws-sdk-php suggests installing ext-apc (Allows service description opcode caching, requ
st and response caching, and credentials caching)
aws/aws-sdk-php suggests installing monolog/monolog (Adds support for logging HTTP requests a
nd responses)
aws/aws-sdk-php suggests installing symfony/yaml (Eases the ability to write manifests for cr
eating jobs in AWS Import/Export)
Package guzzle/guzzle is abandoned, you should avoid using it. Use guzzlehttp/guzzle instead.
Writing lock file
Generating autoload files
[ec2-user@ip-172-31-85-248 face]$
```

2. Installing php

```
ec2-user@ip-172-31-85-248:/var/www/html/face
login as: ec2-user
Authenticating with public key "imported-openssh-key"

      _ _ _ _ _
     /   _ _   \
    / _ _ _ _ _ \
   /               \
  /                 \
 /                   \
/                     \
_ _ _ _ _

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-85-248 ~]$ sudo yum install php
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package php.x86_64 0:5.4.16-46.amzn2.0.2 will be installed
--> Processing Dependency: php-cli(x86-64) = 5.4.16-46.amzn2.0.2 for package: php-5.4.16-46.a
mzn2.0.2.x86_64
--> Processing Dependency: php-common(x86-64) = 5.4.16-46.amzn2.0.2 for package: php-5.4.16-4
6.amzn2.0.2.x86_64
--> Processing Dependency: httpd-mmn = 20120211x8664 for package: php-5.4.16-46.amzn2.0.2.x86
_64
--> Processing Dependency: httpd for package: php-5.4.16-46.amzn2.0.2.x86_64
--> Running transaction check
--> Package httpd.x86_64 0:2.4.41-1.amzn2.0.1 will be installed
--> Processing Dependency: httpd-tools = 2.4.41-1.amzn2.0.1 for package: httpd-2.4.41-1.amzn2
.0.1.x86_64
--> Processing Dependency: httpd-filesystem = 2.4.41-1.amzn2.0.1 for package: httpd-2.4.41-1.
amzn2.0.1.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.41-1.amzn2.0.1.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.41-1.amzn2.0.1.x86_64
--> Processing Dependency: httpd-filesystem for package: httpd-2.4.41-1.amzn2.0.1.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.41-1.amzn2.0.1.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.41-1.amzn2.0.1.x
86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.41-1.amzn2.0.1.x86_6
4
--> Package php-cli.x86_64 0:5.4.16-46.amzn2.0.2 will be installed
--> Package php-common.x86_64 0:5.4.16-46.amzn2.0.2 will be installed
--> Processing Dependency: libzip.so.2()(64bit) for package: php-common-5.4.16-46.amzn2.0.2.x
86_64
--> Running transaction check
--> Package apr.x86_64 0:1.6.3-5.amzn2.0.2 will be installed
--> Package apr-util.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.1-5.amzn2.0.2 for package: apr-util-1.6
.1-5.amzn2.0.2.x86_64
--> Package generic-logos-httpd.noarch 0:18.0.0-4.amzn2 will be installed
--> Package httpd-filesystem.noarch 0:2.4.41-1.amzn2.0.1 will be installed
--> Package httpd-tools.x86_64 0:2.4.41-1.amzn2.0.1 will be installed
--> Package libzip010-compat.x86_64 0:0.10.1-9.amzn2.0.5 will be installed
--> Package mailcap.noarch 0:2.1.41-2.amzn2 will be installed
```

3. index.php file code

```
ec2-user@ip-172-31-85-248:/var/www/html/face
use Aws\S3\S3Client;
use Aws\Rekognition\RekognitionClient;

$bucket = 'aj-reckognition';
$keyname = 's.jpg';

$s3 = new S3Client([
    'region' => 'us-east-1',
    'version' => '2006-03-01',
    'signature' => 'v4'
]);

try {
    // Upload data.
    $result = $s3->putObject([
        'Bucket' => $bucket,
        'Key' => $keyname,
        'SourceFile' => __DIR__ . "/$keyname",
        'ACL' => 'public-read-write'
    ]);

    // Print the URL to the object.
    $imageUrl = $result['ObjectURL'];
    if($imageUrl) {
        echo "Image upload done... Here is the URL: " . $imageUrl;

        $rekognition = new RekognitionClient([
            'region' => 'us-east-1',
            'version' => 'latest',
        ]);

        $result = $rekognition->detectFaces([
            'Attributes' => ['DEFAULT'],
            'Image' => [
                'S3Object' => [
                    'Bucket' => $bucket,
                    'Name' => $keyname,
                    'Key' => $keyname,
                ],
            ],
        ]);

        echo "Totally there are " . count($result["FaceDetails"]) . " faces";
    }
} catch (Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}

"index.php" 82L, 1829C
```

4. Upload success screenshot

```
ec2-user@ip-172-31-85-248:/var/www/html/face
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Sat Apr  4 16:36:01 2020 from 183.82.152.207

 _ _ _ _ _
| | | | |
|_|_|_|_|_| Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-85-248 ~]$ cd /var/www/html
[ec2-user@ip-172-31-85-248 html]$ cd face
[ec2-user@ip-172-31-85-248 face]$ ls
composer.json composer.lock index.php s.jpg vendor
[ec2-user@ip-172-31-85-248 face]$ sudo rm index.php
[ec2-user@ip-172-31-85-248 face]$ ls
composer.json composer.lock s.jpg vendor
[ec2-user@ip-172-31-85-248 face]$ sudo vim index.php
[ec2-user@ip-172-31-85-248 face]$ sudo php index.php
Image upload done... Here is the URL: https://aj-reckognition.s3.amazonaws.com/s
[ec2-user@ip-172-31-85-248 face]$
```

Screenshots needed for EC2 & Rekognition

1. Face Detect success screenshot

```
ec2-user@ip-172-31-85-248:/var/www/html/face
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Sat Apr  4 16:47:02 2020 from 183.82.152.207

 _ _ _ _ _
| | | | |
|_|_|_|_|_| Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-85-248 ~]$ sudo php index.php
Could not open input file: index.php
[ec2-user@ip-172-31-85-248 ~]$ cd /var/www/html
[ec2-user@ip-172-31-85-248 html]$ cd face
[ec2-user@ip-172-31-85-248 face]$ sudo php index.php
Image upload done... Here is the URL: https://aj-rekognition.s3.amazonaws.com/s
[ec2-user@ip-172-31-85-248 face]$ cd face
-bash: cd: face: No such file or directory
[ec2-user@ip-172-31-85-248 face]$ sudo php index.php
Image upload done... Here is the URL: https://aj-rekognition.s3.amazonaws.com/s.jpgTotally there are 9 faces[ec2-user@ip-172-31-85-248 face]$
```