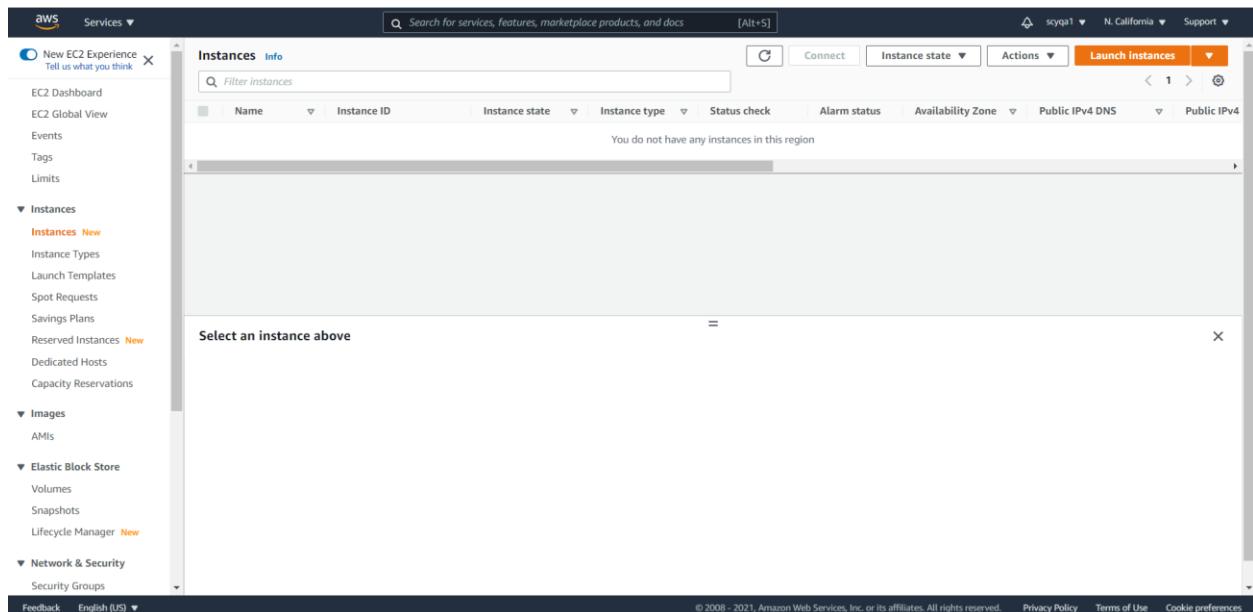


PART I: CI/CD With AWS ECS and Docker [100 points]

1) Docker Basics: Create EC2 Instance. Install Docker. Pull centos:centos6. Create index.html with your greeting. Create Dockerfile. Build a Docker image



Launch an instance



Choose AMI



Deep Learning AMI (Amazon Linux 2) Version 52.0 - ami-0d542c137d9d4b840
MXNet-1.8.0 & 1.7.0, TensorFlow-2.4.3, 2.3.4 & 1.15.5, PyTorch-1.7.1 & 1.8.1, Neuron, & others. NVIDIA CUDA, cuDNN, NCCL, Intel MKL-DNN, Docker, NVIDIA-Docker & EFA support. For fully managed experience, check: <https://aws.amazon.com/sagemaker>

Select

Choose an instance type

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 families ▾ Current generation ▾ Show/Hide Columns

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, ~ 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro <small>(Free tier eligible)</small>	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
<input type="checkbox"/>	t3	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	t3	t3.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	t3	t3.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

Feedback English (US) ▾ © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Configure instance details

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of Instances Launch into Auto Scaling Group

Purchasing option Request Spot instances

Network Subnet Auto-assign Public IP

Placement group Add instance to placement group Capacity Reservation

Domain join directory IAM role

Shutdown behavior Stop - Hibernate behavior Enable hibernation as an additional stop behavior

Enable termination protection Protect against accidental termination

Monitoring Enable CloudWatch detailed monitoring Additional charges apply

Tenancy Additional charges will apply for dedicated tenancy

Cancel Previous Review and Launch Next: Add Storage

Feedback English (US) ▾ © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Add storage

The screenshot shows the 'Add storage' step of the AWS EC2 instance creation wizard. The top navigation bar includes 'Services ▾', a search bar, and account information ('scyqa1 N. California Support'). Below the navigation is a progress bar with steps 1-7: Choose AMI, Choose Instance Type, Configure Instance, Add Storage (highlighted in orange), Add Tags, Configure Security Group, and 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-036cd1d5a6a94ee6	110	General Purpose SSD (gp2)	330 / 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.

Buttons at the bottom: Cancel, Previous, **Review and Launch**, Next: Add Tags.

Footer links: Feedback, English (US) ▾, © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved., Privacy Policy, Terms of Use, Cookie preferences.

Add tags

The screenshot shows the 'Add tags' step of the AWS EC2 instance creation wizard. The top navigation bar and progress bar are identical to the previous screenshot.

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	(128 characters maximum)	Value	(256 characters maximum)	Instances	Volumes	Network Interfaces
This resource currently has no tags						

Choose the Add tag button or click to add a Name tag.
Make sure your IAM policy includes permissions to create tags.

Add Tag (Up to 50 tags maximum)

Buttons at the bottom: Cancel, Previous, **Review and Launch**, Next: Configure Security Group.

Footer links: Feedback, English (US) ▾, © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved., Privacy Policy, Terms of Use, Cookie preferences.

Configure security group

Screenshot of the AWS Step 6: Configure Security Group wizard.

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: Docker

Description: launch-wizard-3 created 2021-10-22T01:48:04.809-07:00

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	My IP	73.202.58.173/32
HTTP	TCP	80	Custom	0.0.0.0/:/0

[Add Rule](#)

[Cancel](#) [Previous](#) [Review and Launch](#)

Review instance launch

Screenshot of the AWS Step 7: Review Instance Launch wizard.

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

Deep Learning AMI (Amazon Linux 2) Version 52.0 - ami-0d542c137d9d4b840

MXNet-1.8.0 & 1.7.0, TensorFlow-2.4.3, 2.3.4 & 1.15.5, PyTorch-1.7.1 & 1.8.1, Neuron, & others. NVIDIA CUDA, cuDNN, NCCL, Intel MKL-DNN, Docker, NVIDIA-Docker & EFA support. For fully managed experience, check: <https://aws.amazon.com/sagemaker>

Root Device Type: ebs Virtualization type: hvm

[Edit AMI](#)

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
I2 micro	-	1	1	EBS only	-	Low to Moderate

[Edit instance type](#)

Security Groups

Security group name: Docker

Description: launch-wizard-3 created 2021-10-22T01:48:04.809-07:00

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	73.202.58.173/32	

[Edit security groups](#)

[Cancel](#) [Previous](#) [Launch](#)

Create a new key pair

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance. Amazon EC2 supports ED25519 and RSA key pair types.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

Key pair type
 RSA ED25519

Key pair name
Docker

Download Key Pair

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel Launch Instances

Now, the instance is created

AWS Services ▾

Instances (1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
-	i-090096fa/e4cd9755	Running	t2.micro	-	No alarms	us-west-1a	ec2-18-144-24-211.us...	18.144.24.2

Select an instance above

Feedback English (US) ▾

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Allocate elastic IP address

The screenshot shows the AWS Elastic IP addresses page. The left sidebar includes sections for Instances, Images, Elastic Block Store, Network & Security (with a highlighted Elastic IPs link), and Load Balancing. The main content area displays a table titled "Elastic IP addresses" with columns: Name, Allocated IPv4 add..., Type, Allocation ID, Reverse DNS record, and Associated instance. A search bar at the top allows filtering by "Filter Elastic IP addresses". A prominent message at the bottom of the table states "No Elastic IP addresses found in this Region".

The screenshot shows the AWS Elastic IP addresses page after allocation. A green banner at the top indicates "Elastic IP address allocated successfully." The main content area displays a table titled "Elastic IP addresses (1/1)" with one entry: a Public IPv4 address of 54.151.31.25, which is a Public IP type with an allocation ID of eipalloc-8e056293. The left sidebar remains the same as the previous screenshot.

Associate elastic IP address

The screenshot shows the AWS Elastic IP Addresses console. At the top, there is a navigation bar with 'Actions' and an orange button labeled 'Allocate Elastic'. Below this, a sidebar lists several actions: 'View details', 'Release Elastic IP addresses', 'Associate Elastic IP address' (which is highlighted in blue), 'Disassociate Elastic IP address', and 'Update reverse DNS'. A green success message at the bottom of the page states: 'Elastic IP address associated successfully. Elastic IP address 54.151.31.25 has been associated with instance i-090096fa7e4cd9755'. The main content area displays a table of elastic IP addresses, with one entry visible: Public IPv4 address: 54.151.31.25, Type: Public IP, Allocation ID: eipalloc-8e056293, Associated instance ID: i-090096fa7e4cd9755, and Private IP address: 172.31.15.27.

SSH connection to EC2

```
aqc@ubuntu:~/aqc$ chmod 400 DockerProject.pem
aqc@ubuntu:~/aqc$ ssh -i DockerProject.pem ec2-user@54.241.47.129
The authenticity of host '54.241.47.129 (54.241.47.129)' can't be established
ECDSA key fingerprint is SHA256:ho/h8SrUPN8yckNS2SDA7JW5tfDz8GAuYT062gBLjBc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.241.47.129' (ECDSA) to the list of known hosts

      _\   _ ) 
     _\  ( _ /   Amazon Linux 2 AMI
    ___\_\_|\__| 

https://aws.amazon.com/amazon-linux-2/
3 package(s) needed for security, out of 15 available
Run "sudo yum update" to apply all updates.

[ec2-user@ip-172-31-10-156 ~]$ sudo -i
[root@ip-172-31-10-156 ~]# yum update
```

Install Docker

```
[root@ip-172-31-10-156 ~]# yum install docker -y
```

```
[root@ip-172-31-10-156 ~]# service docker start
```

Pull centos:centos6

```
[root@ip-172-31-10-156 ~]# docker pull centos:centos6
```

Docker images

```
[root@ip-172-31-10-156 ~]# docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
centos          centos6  5bf9684f4720  5 weeks ago  194MB
```

Docker run

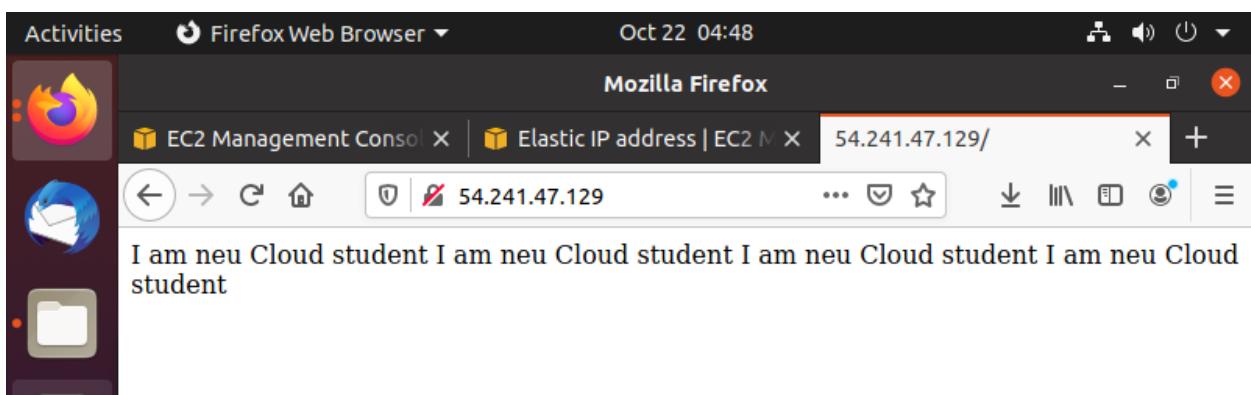
```
[root@ip-172-31-10-156 ~]# docker run -it -p 80:80 5bf9684f4720
[root@431a0adf7580 /]#
```

Install httpd

```
[root@431a0adf7580 /]# yum install httpd
```

Create index.html with my greeting

```
[root@431a0adf7580 /]# cd /var/www/html
[root@431a0adf7580 html]# echo "I am neu Cloud student" >> index.html
[root@431a0adf7580 html]# service httpd start
Starting httpd: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2 for ServerName
[ OK ]
```



Create Dockerfile.



A screenshot of a terminal window titled "root@ip-172-31-10-156:~/docker". The terminal contains the following Dockerfile code:

```
FROM centos:centos6
MAINTAINER VarumMnaik
RUN yum -y install httpd
COPY index.html /var/www/html/
CMD ["/usr/sbin/httpd", "-D", "FOREGROUND"]
EXPOSE 80
```

2) [17:57] AWS ECR(Elastics Container Registry) Creation: Create AWS ECR. Login to ECR. Tag existing image as AWS ECR repo. Push the image into the ECR

Create ECR

The screenshot shows the AWS ECR (Amazon Elastic Container Registry) service in the AWS Management Console. The left sidebar lists services like Amazon ECS, Amazon EKS, and Amazon ECR. The main content area shows a list of private repositories under the 'Private' tab. A success message at the top says 'Successfully created repository scyqa1'. The repository 'scyqa1' is listed with details: Repository name: scyqa1, URI: 008889205193.dkr.ecr.us-west-1.amazonaws.com/scyqa1, Created at: 2021年10月22日, 07:01:40 (UTC-07), Tag immutability: Disabled, Scan on push: Disabled, Encryption type: AES-256.

Create a role

The screenshot shows the AWS IAM (Identity and Access Management) service in the AWS Management Console. The left sidebar lists categories like Access management, Policies, and Access reports. The main content area shows a list of roles under the 'Roles' tab. A search bar at the top right shows 'Roles (10) Info'. The list includes roles such as '1-dev-ZappaLambdaExecutionRole', '1-commerce-ZappaLambdaExecutionRole', 'AWSServiceRoleForAPIGateway', 'AWSServiceRoleForAutoScaling', 'AWSServiceRoleForElasticLoadBalancing', 'AWSServiceRoleForSupport', 'AWSServiceRoleForTrustedAdvisor', 'django-dev-ZappaLambdaExecutionRole', 'EC2_access_project2', and 'HelloWorld-role-milm13seh'. Each role entry shows its last activity date.

Modify EC2's IAM role

The screenshot shows the AWS EC2 Instances page. A context menu is open over an instance named 'i-097b24608f302b5bd'. The 'Security' submenu is expanded, and the 'Modify IAM role' option is selected. The main table lists one instance: 'i-097b24608f302b5bd' (Running, t2.micro, 2/2 checks passed, us-west-1a, Public IPv4 DNS: ec2-54-241-47-129.us...). Below the table, two tabs are visible: 'Inbound rules' and 'Outbound rules'.

Retrieve an authentication token and authenticate the Docker client to registry using the AWS CLI:

```
[root@ip-172-31-10-156 docker]# aws ecr get-login-password --region us-west-1 | docker login --username AWS --password-stdin 008889205193.dkr.ecr.us-west-1.amazonaws.com
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

After the build completes, tag the image

```
[root@ip-172-31-10-156 docker]# docker tag centos:centos6 008889205193.dkr.ecr.us-west-1.amazonaws.com/scyqa1:latest
```

push the image to newly created AWS repository

```
[root@ip-172-31-10-156 docker]# docker tag centos:centos6 008889205193.dkr.ecr.us-west-1.amazonaws.com/scyqa1:latest
Error response from daemon: No such image: centos:centos6
[root@ip-172-31-10-156 docker]# docker tag centos:centos6 008889205193.dkr.ecr.us-west-1.amazonaws.com/scyqa1:latest
[root@ip-172-31-10-156 docker]# docker push 008889205193.dkr.ecr.us-west-1.amazonaws.com/scyqa1:latest
The push refers to repository [008889205193.dkr.ecr.us-west-1.amazonaws.com/scyqa1]
af6fb1987c2e: Pushed
latest: digest: sha256:3688aa867eb84332460e172b9250c9c198fdfd8d987605fd53f246f498c60bcf size: 529
```

3) Create ALB(Application Load Balancer): Create Target group. Create ALB

Create target group

The screenshot shows the 'Specify group details' step of the 'Create target group' wizard in the AWS EC2 console. The left sidebar shows 'Step 1 Specify group details' and 'Step 2 Register targets'. The main area is titled 'Specify group details' with the sub-section 'Basic configuration'. It states: 'Your load balancer routes requests to the targets in a target group and performs health checks on the targets.' Below this is a section titled 'Choose a target type' with four options: 'Instances' (selected), 'IP addresses', 'Lambda function', and 'Application Load Balancer'. The 'Instances' option is described as supporting load balancing to instances within a specific VPC. The other three options have their descriptions collapsed. Below this is a 'Target group name' field containing 'DockerProject2', with a note that names must be alphanumeric and cannot start or end with a hyphen. At the bottom are 'Protocol' (HTTP) and 'Port' (80) settings, and a 'VPC' section.

Search for services, features, marketplace products, and docs [Alt+S]

scyqa1 N. California Support

EC2 > Target groups > Create target group

Step 1
Specify group details

Step 2
Register targets

Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration

Settings in this section cannot be changed after the target group is created.

Choose a target type

Instances

- Supports load balancing to instances within a specific VPC.

IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.

Lambda function

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

Application Load Balancer

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

DockerProject2

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol Port

HTTP : 80

VPC

Feedback English (US) © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Create application load balancer

The screenshot shows the 'Create Application Load Balancer' wizard on the AWS Management Console. The 'Basic configuration' step is selected. It includes fields for 'Load balancer name' (set to 'Docker'), 'Scheme' (set to 'Internet-facing'), 'IP address type' (set to 'IPv4'), and 'Network mapping' (which lists subnets). The 'VPC' tab is also visible.

Load balancer name
Name must be unique within your AWS account and cannot be changed after the load balancer is created.

Scheme [Info](#)
Scheme cannot be changed after the load balancer is created.

Internet-facing
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

Internal
An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type [Info](#)
Select the type of IP addresses that your subnets use.

IPv4
Recommended for internal load balancers.

Dualstack
Includes IPv4 and IPv6 addresses.

Network mapping [Info](#)
The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [Info](#)

The screenshot shows the EC2 Instances page. A new load balancer named 'DockerProject2' is being created. The 'Basic Configuration' section shows the name 'DockerProject2' and other settings. The left sidebar shows navigation links for EC2 services like Dashboard, Global View, and Instances.

Create Load Balancer Actions ▾

Name	DNS name	State	VPC ID	Availability Zones	Type	Created
DockerProject2	DockerProject2-1404174096...	Provisioning	vpc-c57ad0a3	us-west-1b, us-west-1a	application	October 2

Load balancer: DockerProject2

Description **Listeners** **Monitoring** **Integrated services** **Tags**

Basic Configuration

Name: DockerProject2

Feedback English (US) ▾

4) Create an AWS ECS(Elastics Container Service) Cluster: Create a Task with Fargate Computability. Create a Fargate Cluster. Create a Service with ALB enabled. Test ALB DNS, whether your site is running or not. Add your container with auto-config CloudWatch Logs

Create new Task definition

Select launch type compatibility

Select which launch type you want your task definition to be compatible with based on where you want to launch your task.

- FARGATE**
 - Icon: A yellow hat with stars.
 - Description: Price based on task size, Requires network mode awsvpc, AWS-managed infrastructure, no Amazon EC2 instances to manage.
- EC2**
 - Icon: Three yellow squares.
 - Description: Price based on resource usage, Multiple network modes available, Self-managed infrastructure using Amazon EC2 instances.
- EXTERNAL**
 - Icon: A globe icon.
 - Description: Price based on instance-hours and additional charges for other AWS services used, Self-managed on-premise infrastructure with ECS.

Add container

Add container

Standard

Container name* web1

Image* 008889205193.dkr.ecr.us-west-1.amazonaws.com/scyqa1:latest

Private repository authentication*

Memory Limits (MiB) Soft limit 128

Add Hard limit

Define hard and/or soft memory limits in MiB for your container. Hard and soft limits correspond to the 'memory' and 'memoryReservation' parameters, respectively, in task definitions. ECS recommends 300-500 MiB as a starting point for web applications.

Port mappings

Container port	Protocol
80	tcp

Add port mapping

Host port mappings are not valid when the network mode for a task definition is host or awsvpc. To specify different host and container port mappings, choose the Bridge network mode.

* Required

Cancel Add

Log configuration Auto-configure CloudWatch Logs

Log driver: awslogs

Log options:

Key	Value	
awslogs-group	/ecs/cicd	X
awslogs-region	us-west-1	X
awslogs-stream-prefix	ecs	X
Add key	Value	Add value

AWS Services Search for services, features, marketplace products, and docs [Alt+S] scyqa1 N California Support

Launch Status

Task definition status - 3 of 3 completed

Create Execution Role

Execution Role AmazonECSTaskExecutionRole created [Learn more](#)

Create Task Definition: cicd

cicd succeeded

Create CloudWatch Log Group

CloudWatch Log Group created
CloudWatch Log Group /ecs/cicd

Back [View task definition](#)

Create cluster

AWS Services Search for services, features, marketplace products, and docs [Alt+S] scyqa1 N California Support

Create Cluster

Step 1: Select cluster template

Step 2: Configure cluster

Select cluster template

The following cluster templates are available to simplify cluster creation. Additional configuration and integrations can be added later.

Networking only Resources to be created: Cluster VPC (optional) Subnets (optional) <small>For use with either AWS Fargate or External instance capacity.</small>	EC2 Linux + Networking Resources to be created: Cluster VPC Subnets Auto Scaling group with Linux AMI
EC2 Windows + Networking Resources to be created: Cluster VPC Subnets Auto Scaling group with Windows AMI	

Feedback English (US) ▾ © 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Create service

The screenshot shows the AWS ECS Cluster details page for 'cicd-cluster'. The cluster ARN is arn:aws:ecs:us-west-1:008889205193:cluster/cicd-cluster, and its status is PROVISIONING. There are no registered container instances, pending tasks, or running tasks. The active service count is 0, and there are no draining services. The Services tab is selected, showing a table with columns: Service Name, Status, Service type, Task Definition ..., Desired tasks ..., Running tasks ..., Launch type, and Platform versio... . The table is empty, indicating 'No results'.

The screenshot shows the 'Create Service' wizard, Step 1: Configure service. It includes a sidebar with steps: Step 1: Configure service (selected), Step 2: Configure network, Step 3: Set Auto Scaling (optional), and Step 4: Review. The main content area is titled 'Configure service' and describes how a service specifies task definitions and optional load balancing. It shows configuration for Launch type (set to FARGATE), Task Definition (Family: cicd, Revision: 1 (latest)), Platform version (LATEST), Cluster (cicd-cluster), and Service name (serviceProject2).

Create Service

Step 1: Configure service
Step 2: Configure network
Step 3: Set Auto Scaling (optional)
Step 4: Review

Configure network

VPC and security groups

VPC and security groups are configurable when your task definition uses the awsvpc network mode.

Cluster VPC: vpc-c57ad0a3 (172.31.0.0/16)

Subnets:

- subnet-9dfcddc7 (172.31.0.0/20) - us-west-1a assign ipv6 on creation: Disabled
- subnet-6ec1a408 (172.31.16.0/20) - us-west-1b assign ipv6 on creation: Disabled

Security groups: servic-1753

Auto-assign public IP: ENABLED

Health check grace period

If your service's tasks take a while to start and respond to ELB health checks, you can specify a health check grace period of up to 2,147,483,647 seconds during which the ECS service scheduler will ignore ELB health check status. This grace period can prevent the ECS service scheduler from marking tasks as unhealthy and stopping them before they have time to come up. This is only valid if your service is configured to use a load balancer.

Feedback English (US) © 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Launch Status

ECS Service status - 5 of 5 completed

Configure Task Networking

Create security group

- ✓ Create security group succeeded sg-03dccc4d284164cb

Set inbound rules

- ✓ Set inbound rules succeeded sg-03dccc4d284164cb

Create Load Balancer

Target Group: ecs-cicd-c-serviceProject2

- ✓ Target Group created Target Group created. Waiting to create listener/rule. View: ecs-cicd-c-serviceProject2

Rule: 80:HTTP /*:1

- ✓ Rule created

Feedback English (US) © 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

DNS Test

dockerproject2-1404174096.us-west-1.elb.amazonaws.com

5) [38:00] AWS Code commit: Create a Repo. Set ssh connectivity in your local machine. Push your code into the newly created repo

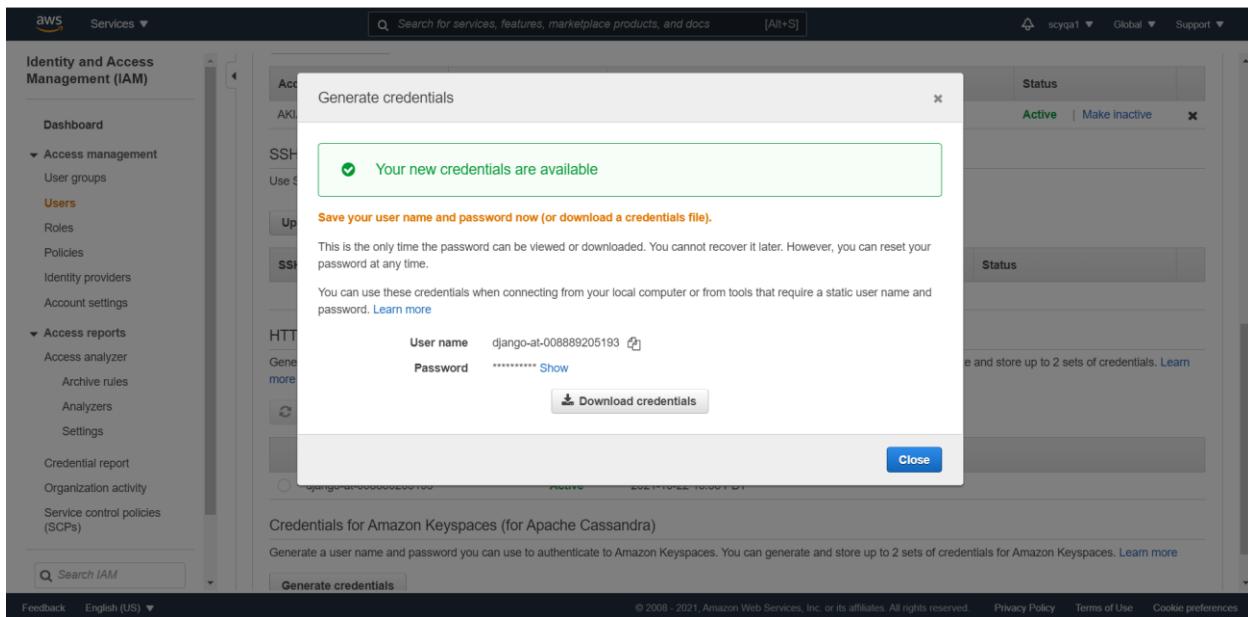
Create a repo

The screenshot shows the 'Create repository' page in the AWS CodeCommit service. The top navigation bar includes the AWS logo, 'Services' dropdown, search bar ('Search for services, features, marketplace prc [Alt+S]'), user account ('scyqa1'), region ('N. California'), and support links. The breadcrumb trail shows 'Developer Tools > CodeCommit > Repositories > Create repository'. The main section is titled 'Create repository' with the sub-section 'Repository settings'. It contains fields for 'Repository name' (set to 'docker-repo'), 'Description - optional' (empty), and 'Tags' (with an 'Add' button). At the bottom are 'Cancel' and 'Create' buttons. The footer includes links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences', along with a copyright notice: '© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.'

Install git

```
[ec2-user@ip-172-31-10-156 docker]$ yum install git
```

Generate HTTPS Git credentials for AWS CodeCommit



Clone

```
[root@ip-172-31-10-156 docker]# git clone https://git-codecommit.us-west-1.amazonaws.com/v1/repos/docker-repo
Cloning into 'docker-repo'...
Username for 'https://git-codecommit.us-west-1.amazonaws.com': django-at-008889205193
Password for 'https://django-at-008889205193@git-codecommit.us-west-1.amazonaws.com':
warning: You appear to have cloned an empty repository.
```

Generate SSH key

```
[root@ip-172-31-10-156 docker]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:wCVWeA8py79YJ2CZKLhn8GhzkVA0joEsnWjY+ND03as root@ip-172-31-10-156.us-west-1.compute.internal
The key's randomart image is:
+---[RSA 2048]---+
|+X==. ooo.      |
|Bo0+.+o++       |
|o= .oo+*.o      |
|o o.. B. ..     |
| =o. . OS       |
|oo+ .+ .        |
|.o   Eo +       |
|
```

Then upload ssh key

The screenshot shows the AWS Lambda SSH keys configuration interface. At the top, there is a button labeled "Upload SSH public key". Below it, a table lists an uploaded SSH key. The table has two columns: "SSH key ID" and "Uploaded". The first row shows the ID "APKAQEEOWQHEZXTHRCFI" and the date "2021-10-2". There is also a link "Show SSH key" next to the ID.

Configure

A terminal window titled "root@ip-172-31-10-156:~/ssh" displays the following SSH configuration:

```
Host git-codecommit.*.amazonaws.com
User APKAQEEOWQHEZXTHRCFI
IdentityFile ~/.ssh/id_rsa
```

Git operations to push

```
[root@ip-172-31-10-156 docker-repo]# git add .
[root@ip-172-31-10-156 docker-repo]# git commit -am"Primary Files"
[master (root-commit) 9b4ce6e] Primary Files
Committer: root <root@ip-172-31-10-156.us-west-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

git config --global --edit

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

2 files changed, 9 insertions(+)
create mode 100644 Dockerfile
create mode 100644 index.html
[root@ip-172-31-10-156 docker-repo]# git push
```

AWS Services ▾

Developer Tools > CodeCommit > Repositories > docker-repo

docker-repo

Name: Dockerfile, index.html

Add file ▾

Feedback English (US) ▾

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

This screenshot shows the AWS CodeCommit interface. On the left, there's a navigation sidebar with sections like 'Source', 'Code', 'Pull requests', 'Commits', 'Branches', 'Git tags', 'Settings', 'Approval rule templates', 'Build', 'Deploy', 'Pipeline', and 'Settings'. The 'Code' section is currently selected. The main area displays a repository named 'docker-repo'. It lists two files: 'Dockerfile' and 'index.html'. There are buttons for 'Notify' (dropdown), 'master' (dropdown), 'Create pull request', 'Clone URL' (dropdown), and 'Add file' (dropdown). At the bottom, there are links for 'Feedback', language selection ('English (US)'), and legal notices ('© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.', 'Privacy Policy', 'Terms of Use', 'Cookie preferences').

6) [54:00] AWS Code Build: Create a Code build Project. Select an Artifact for output. Build your Docker image & push to AWS ECR

Create build project

The screenshot shows two consecutive screenshots of the AWS CodeBuild 'Create build project' interface.

Screenshot 1: Create build project - Project configuration

- Project name:** docker-build
- Description - optional:** (empty)
- Build badge - optional:** Enable build badge
- Enable concurrent build limit - optional:** Limit the number of allowed concurrent builds for this project.
 Restrict number of concurrent builds this project can start
- Additional configuration tags:** (empty)

Screenshot 2: Project created - docker-build

A green banner at the top says: "Project created You have successfully created the following project: docker-build".

Configuration:

Source provider: AWS CodeCommit	Primary repository: docker-repo	Artifacts upload location: -	Build badge: Disabled
Public builds: Disabled			

Build history: No results. There are no results to display.

Configure

```
root@ip-172-31-10-156:/opt/docker/docker-repo
version 0.2

phases:
  pre_build:
    commands:
      -echo Logging in to Amazon ECR...
      -$(aws ecr get-login --no-include-email --region $AWS_DEFAULT_REGION)
  build:
    commands:
      -echo Build started on `date`
      -echo Building the Docker image...
      -docker build -t web:1 .
      -docker tag web:1 008889205193.dkr.ecr.us-west-1.amazonaws.com/scyqa1
  post_build:
    commands:
      -echo Build completed on `date`
      -echo Pushing the Docker image...
      -docker push 008889205193.dkr.ecr.us-west-1.amazonaws.com/scyqa1
~
~
~
~
~
"buildspec.yaml" 18L, 716C          18,83          All

[root@ip-172-31-10-156 docker-repo]# git add .
[root@ip-172-31-10-156 docker-repo]# git commit -am"Add buildspec.yaml"
[master 8babf19] Add buildspec.yaml
Committer: root <root@ip-172-31-10-156.us-west-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

git config --global --edit

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

1 file changed, 18 insertions(+)
create mode 100644 buildspec.yaml
[root@ip-172-31-10-156 docker-repo]# git push
```

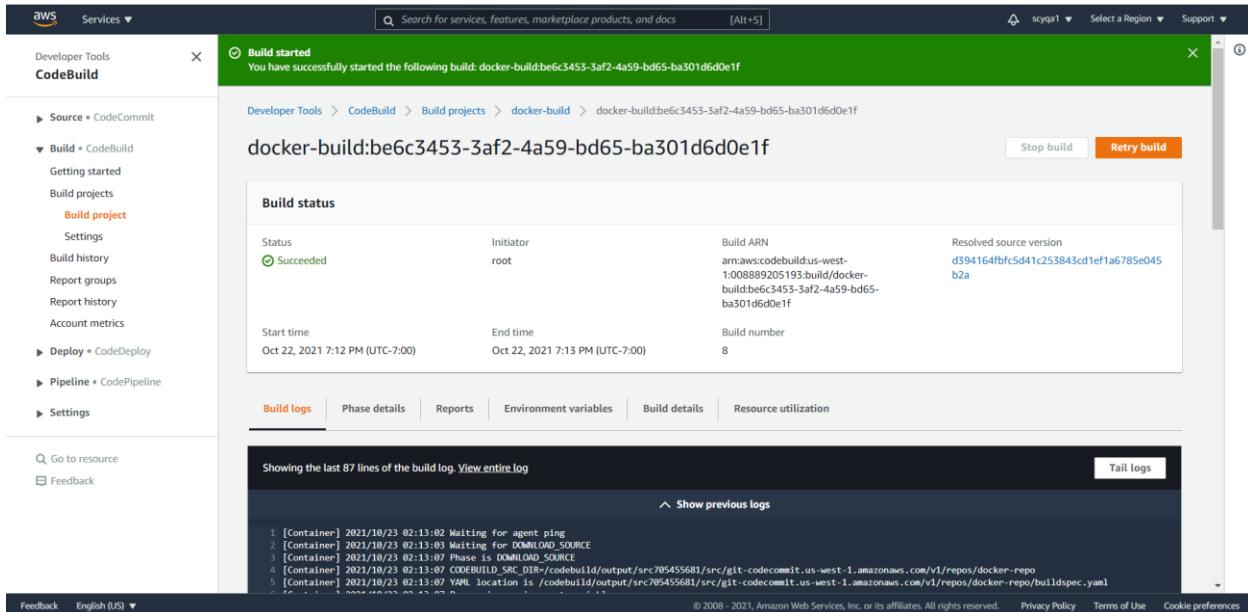
Start build

The screenshot shows the AWS CodeBuild console with a green banner at the top stating "Build started" and "You have successfully started the following build: docker-build:961ed5ef-c282-4ca5-8175-b3eade5ad6c8". The main area displays the build status for "docker-build:961ed5ef-c282-4ca5-8175-b3eade5ad6c8", which is currently "In progress". It provides details like Start time (Oct 22, 2021 6:13 PM UTC-7:00), End time (-), Initiator (root), Build ARN (arn:aws:codebuild:us-west-1:008889205193:build/docker-build:961ed5ef-c282-4ca5-8175-b3eade5ad6c8), and Resolved source version (-). Below the status, there are tabs for "Build logs", "Phase details", "Reports", "Environment variables", "Build details", and "Resource utilization". The "Build logs" tab is selected, showing a message "Showing the last 1000 lines of the build log. View entire log" and a "Tail logs" button. The left sidebar includes sections for Source, Build, Deploy, Pipeline, and Settings, with "Build project" currently selected.

Add policy to roles

The screenshot shows the AWS IAM console under the "Identity and Access Management (IAM)" section. A modal window is open, stating "New feature to generate a policy based on CloudTrail events. AWS uses your CloudTrail events to identify the services and actions used and generate a least privileged policy that you can attach to this role." The main area shows the "Summary" of the "codebuild-docker-build-service-role". It lists the Role ARN (arn:aws:iam::008889205193:role/service-role/codebuild-docker-build-service-role), Role description (Edit), Instance Profile ARNs (Edit), Path (/service-role/), Creation time (2021-10-22 17:21 PDT), Last activity (2021-10-22 18:27 PDT (Today)), and Maximum session duration (1 hour Edit). Below the summary, there are tabs for "Permissions", "Trust relationships", "Tags", "Access Advisor", and "Revoke sessions". The "Permissions" tab is selected, showing a list of applied policies: "Permissions policies (2 policies applied)". Under "Attach policies", there is a table with columns "Policy name" and "Policy type". The table contains two entries: "AmazonEC2ContainerRegistryFullAccess" (AWS managed policy) and "CodeBuildBasePolicy-docker-build-us-west-1" (Managed policy). A "Add inline policy" button is also visible.

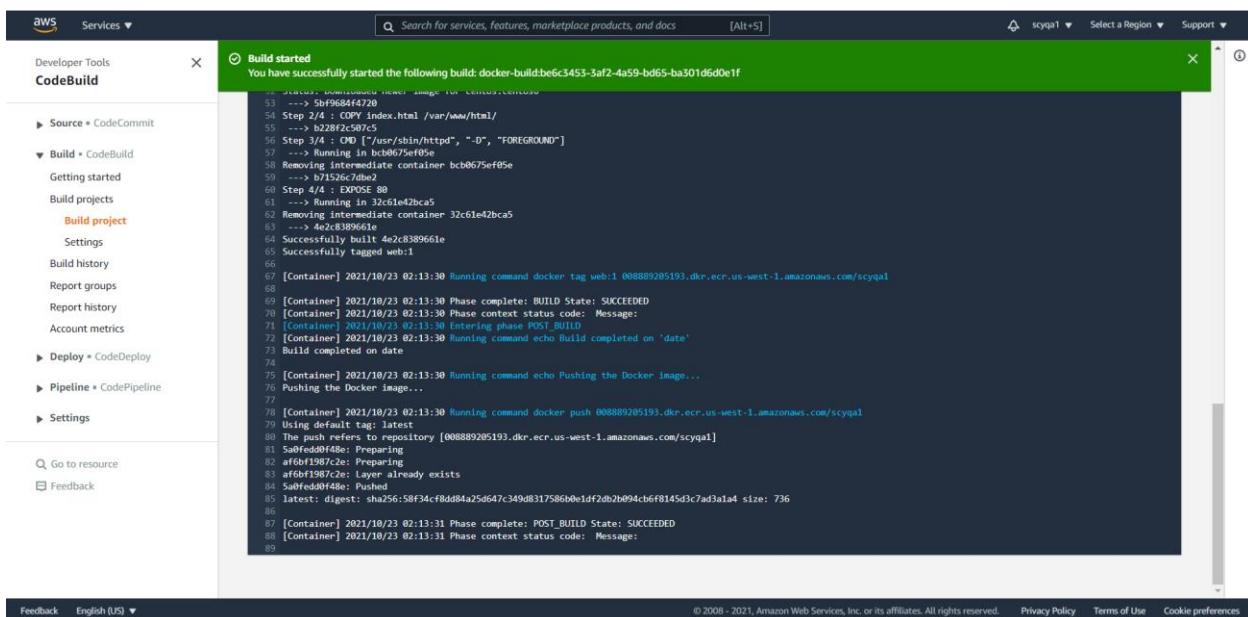
Success



The screenshot shows the AWS CodeBuild console after a successful build. The top navigation bar includes the AWS logo, Services dropdown, search bar, and user account information (scyq1, Select a Region, Support). The main content area has a green header bar stating "Build started" and "You have successfully started the following build: docker-build:be6c3453-3af2-4a59-bd65-ba301d6d0e1f". Below this, the build project name "docker-build:be6c3453-3af2-4a59-bd65-ba301d6d0e1f" is displayed. A "Build status" card provides details: Status "Succeeded", Initiator "root", Build ARN "arn:aws:codebuild:us-west-1:008889205193:build/docker-build:be6c3453-3af2-4a59-bd65-ba301d6d0e1f", and Resolved source version "d394164fbfc5d41c253843cd1ef1a6785e045b2a". It also shows the build duration from Oct 22, 2021 7:12 PM (UTC-7:00) to Oct 22, 2021 7:13 PM (UTC-7:00), and a build number of 8. Below the status card are tabs for "Build logs", "Phase details", "Reports", "Environment variables", "Build details", and "Resource utilization". The "Build logs" tab is selected, showing the last 87 lines of the build log. The log output is as follows:

```
Showing the last 87 lines of the build log. View entire log
Tail logs
^ Show previous logs
1 [Container] 2021/10/23 02:13:02 Waiting for agent ping
2 [Container] 2021/10/23 02:13:02 Phase: DOWNLOAD_SOURCE
3 [Container] 2021/10/23 02:13:07 Phase is DOWNLOAD_SOURCE
4 [Container] 2021/10/23 02:13:07 CODEBUILD_SRC_DIR=/codebuild/output/src/705455681/src/git-codecommit.us-west-1.amazonaws.com/v1/repos/docker-repo
5 [Container] 2021/10/23 02:13:07 YAML location is /codebuild/output/src/705455681/src/git-codecommit.us-west-1.amazonaws.com/v1/repos/docker-repo/buildspec.yaml
```

At the bottom of the page are links for Feedback, English (US), and various AWS terms.



This screenshot shows the same AWS CodeBuild success build log as the first one, but with a much longer log output (lines 53 to 89). The log details the Docker build process, including copying files, running commands like "RUN", "COPY", and "EXPOSE", and finally pushing the built Docker image to Amazon ECR. The log ends with a message indicating the build completed successfully.

```
53 --> 5bf9684f472b
54 Step 2/4 : COPY index.html /var/www/html/
55 --> b228f2c507c5
56 Step 3/4 : CMD ["/usr/sbin/httpd", "-D", "FOREGROUND"]
57 --> Running command bcb8675ef05e
58 Removing intermediate container bcb8675ef05e
59 --> b71526c7d0e2
60 Step 4/4 : EXPOSE 80
61 --> Running in 32c61e42bca5
62 Removing intermediate container 32c61e42bca5
63 --> 4e2c8389661e
64 Successfully built 4e2c8389661e
65 Successfully tagged web:1
66
67 [Container] 2021/10/23 02:13:30 Running command docker tag web:1 008889205193.dkr.ecr.us-west-1.amazonaws.com/scyq1
68
69 [Container] 2021/10/23 02:13:30 Phase complete: BUILD State: SUCCEEDED
70 [Container] 2021/10/23 02:13:30 Phase context status code: Message:
71 [Container] 2021/10/23 02:13:30 Entering phase POST_BUILD
72 [Container] 2021/10/23 02:13:30 Running command echo Build completed on `date`
73 Build completed on date
74
75 [Container] 2021/10/23 02:13:30 Running command echo Pushing the Docker image...
76 Pushing the Docker image...
77
78 [Container] 2021/10/23 02:13:30 Running command docker push 008889205193.dkr.ecr.us-west-1.amazonaws.com/scyq1
79 Using default tag: latest
80 The push refers to repository [008889205193.dkr.ecr.us-west-1.amazonaws.com/scyq1]
81 5a0feddf4f48: Preparing
82 af6b1f967c2e: Preparing
83 5a0feddf4f48: Already exists
84 5a0feddf4f48: Pushed
85 latest: digest: sha256:58f34cf8dd8425d647c349d8317586b0e1df2db2b94cb6f8145d3c7ad3a1a4 size: 736
86
87 [Container] 2021/10/23 02:13:31 Phase complete: POST_BUILD State: SUCCEEDED
88 [Container] 2021/10/23 02:13:31 Phase context status code: Message:
```

At the bottom of the page are links for Feedback, English (US), and various AWS terms.

AWS Services ▾

Search for services, features, marketplace products, and docs [Alt+S]

scyqa1 N. California Support

Amazon Container Services

Amazon ECS Clusters Task definitions

Amazon EKS Clusters

Amazon ECR Repositories

Images

Permissions Lifecycle Policy Tags

Private registry Public registry Public gallery

Amazon ECR > Repositories > scyqa1

scyqa1

View push commands Edit

Images (2)

Find images

<input type="checkbox"/>	Image tag	Pushed at	Size (MB)	Image URI	Digest	Scan status	Vulnerabilities
<input type="checkbox"/>	latest	2021年10月22日, 19:13:32 (UTC-07)	69.84	Copy URI	sha256:58f34cf8dd84a25...	-	-
<input type="checkbox"/>	<untagged>	2021年10月22日, 07:09:48 (UTC-07)	69.84	Copy URI	sha256:3688aa867eb843...	-	-

Feedback English (US) ▾ © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

7) [1:06:55] AWS Code Pipeline (the manual one is optional and you can skip it but not harm to try, automated way starts at [1:10:05]): Create a pipeline Project. Select your Code commit repo as an input repo from step 5. Select Code build project from Step 6. Select Code Deploy for ECS. Select your cluster and Service name from step 3. Create the Code pipeline

Create new revision of task definitions

The screenshot shows the AWS Task Definitions page. On the left, there's a sidebar with links like 'New ECS Experience', 'Clusters', 'Task Definitions' (which is selected and highlighted in orange), 'Account Settings', 'Amazon ECR', 'Repositories', 'AWS Marketplace', 'Discover software', and 'Subscriptions'. The main content area has a title 'Task Definitions' and a sub-instruction: 'Task definitions specify the container information for your application, such as how many containers are part of your task, what resources they will use, how they are linked together, and which host ports they will use.' Below this is a search bar and a status filter: 'Status: ACTIVE INACTIVE 1 selected'. A table lists task definitions: 'Task Definition' (checkbox), 'Latest revision status' (checkbox), and 'Actions' (dropdown). One row is selected: 'cicd' (checkbox checked, status 'ACTIVE'). At the bottom right of the table, there are buttons for '< 1-1 >' and 'Page size 50'. The footer of the page includes links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

Update container

The screenshot shows the 'Edit container' dialog box. On the left, there's a sidebar with sections for 'Task CPU maximum allocation for containers', 'Container definitions' (with a 'Container Name' dropdown set to 'web1'), 'Service integration' (with a note about AWS App Mesh), 'Proxy configuration' (with a note about App Mesh proxy), and 'Log router integration' (with a note about FireLens). The main dialog box has a title 'Edit container' and a section 'Standard'. It contains fields for 'Container name*' (set to 'web1'), 'Image*' (set to '008889205193.dkr.ecr.us-west-1.amazonaws.com/scyqa1/latest'), 'Private repository authentication*' (unchecked), 'Memory Limits (MiB)' (set to 'Soft limit 128'), and 'Port mappings' (Container port 80, Protocol tcp). A note at the bottom of the dialog says: 'Host port mappings are not valid when the network mode for a task definition is host or awsvpc. To specify different host and container port mappings, choose the Bridge network mode.' At the bottom right, there are 'Cancel' and 'Update' buttons.

Screenshot of the AWS ECS Task Definitions console showing the creation of a new task definition revision.

Task Definition: cicd-rev:1

Task definition name: cicd-rev

Task role: ecsTaskExecutionRole

Network mode: awsvpc

Compatibilities: EC2, FARGATE

Requires compatibilities: FARGATE

Actions dropdown menu:

- Run Task
- Create Service
- Update Service

Feedback English (US) © 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Update service

Screenshot of the AWS ECS Task Definitions console showing the list of task definitions.

Task Definitions

Status: ACTIVE (1 selected)

Filter: cicd-rev

Latest revision status: ACTIVE

Actions dropdown menu:

- Run Task
- Create Service
- Update Service

Last updated on October 22, 2021 7:45:03 PM (0m ago)

Feedback English (US) © 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Launch Status

ECS Service status - 1 of 1 completed

Configure Task Networking

Service Auto Scaling

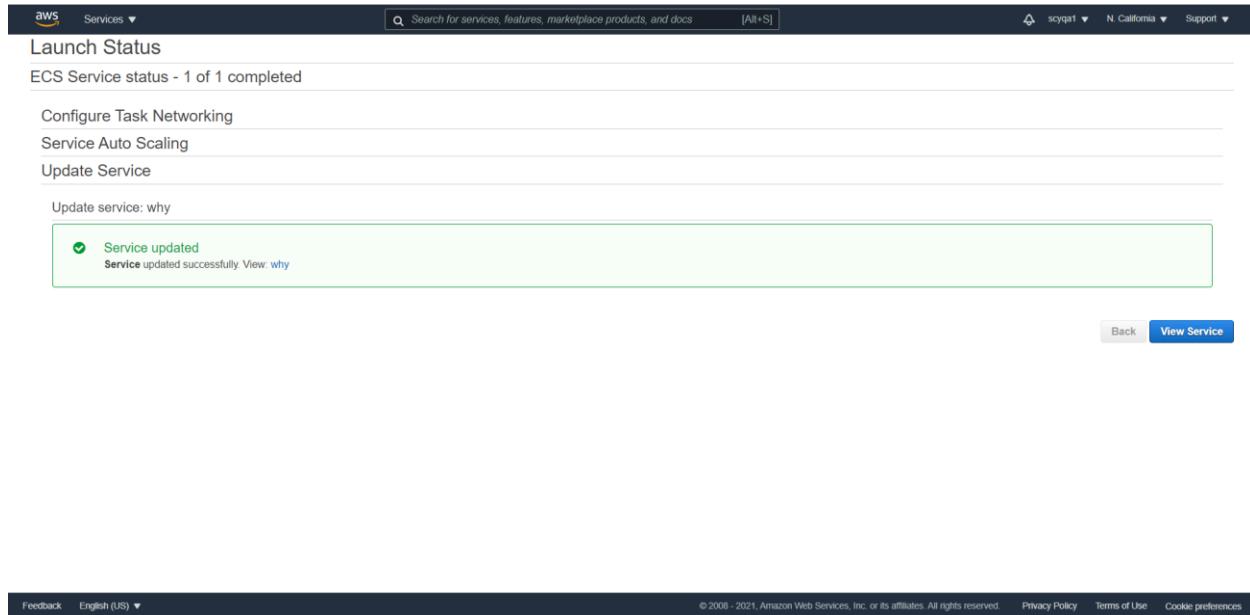
Update Service

Update service: why

Service updated
Service updated successfully. View why

Back View Service

Feedback English (US) © 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences



Create pipeline

Choose pipeline settings

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1 Choose pipeline settings

Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
 No more than 100 characters

Service role

New service role
Create a service role in your account

Existing service role
Choose an existing service role from your account

Role name
 Type your service role name

Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

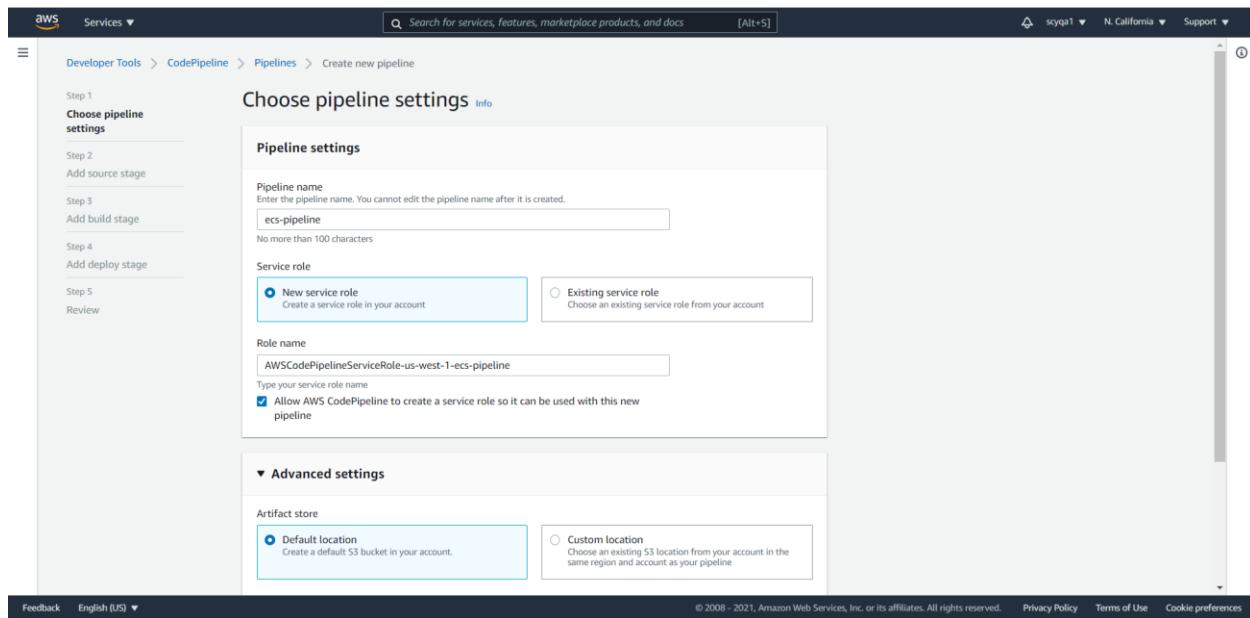
Advanced settings

Artifact store

Default location
Create a default S3 bucket in your account.

Custom location
Choose an existing S3 location from your account in the same region and account as your pipeline

Feedback English (US) © 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences



Add source stage

The screenshot shows the 'Add source stage' configuration screen. On the left, a sidebar lists steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage, currently selected), Step 3 (Add build stage), Step 4 (Add deploy stage), and Step 5 (Review). The main area is titled 'Source'. It includes fields for 'Source provider' (set to 'AWS CodeCommit'), 'Repository name' ('Q docker-repo'), 'Branch name' ('Q master'), and 'Change detection options' (radio button selected for 'Amazon CloudWatch Events (recommended)'). Below these are sections for 'Output artifact format' (radio button selected for 'CodePipeline default') and 'Full clone' (radio button for 'AWS CodePipeline passes metadata about the repository'). At the bottom are standard navigation buttons: 'Cancel', 'Previous', 'Skip build stage', and 'Next'.

Add build stage

The screenshot shows the 'Add build stage' configuration screen. The sidebar shows steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage, currently selected), Step 4 (Add deploy stage), and Step 5 (Review). The main area is titled 'Build - optional'. It includes fields for 'Build provider' (set to 'AWS CodeBuild'), 'Region' ('US West (N. California)'), and 'Project name' ('Q docker-build'). There is also a 'Create project' button. Below these are sections for 'Environment variables - optional' (with a 'Learn more' link) and 'Build type' (radio button selected for 'Single build'). At the bottom are standard navigation buttons: 'Cancel', 'Previous', 'Skip build stage', and 'Next'.

Add deploy stage

The screenshot shows the AWS CodePipeline 'Add deploy stage' configuration interface. The left sidebar lists navigation options: Source (CodeCommit), Build (CodeBuild), Deploy (CodeDeploy), Pipeline (CodePipeline), Getting started, Pipelines, and Settings. The main area shows a step-by-step process: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage, currently selected), and Step 5 (Review). The 'Add deploy stage' step is titled 'Deploy - optional'. It includes fields for 'Deploy provider' (Amazon ECS), 'Region' (US West (N. California)), 'Cluster name' (cicd-cluster), 'Service name' (serviceProject2), 'Image definitions file - optional' (Myfilename.json), and 'Deployment timeout - optional' (empty field).

Developer Tools > CodePipeline > Pipelines > Create new pipeline [Alt+S]

Developer Tools > CodePipeline > Pipelines > Create new pipeline [Alt+S]

Step 1 Choose pipeline settings

Step 2 Add source stage

Step 3 Add build stage

Step 4 Add deploy stage

Step 5 Review

Add deploy stage Info

Deploy - optional

Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

Amazon ECS

Region
US West (N. California)

Cluster name
Choose a cluster that you have already created in the Amazon ECS console. Or create a cluster in the Amazon ECS console and then return to this task.

cicd-cluster

Service name
Choose a service that you have already created in the Amazon ECS console for your cluster. Or create a new service in the Amazon ECS console and then return to this task.

serviceProject2

Image definitions file - optional
Enter the JSON file that describes your service's container name and the image and tag.

Myfilename.json

Deployment timeout - optional
Enter the timeout in minutes for the deployment action.

Feedback English (US) © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

8) [1:19:45] Error & Resolution: In the above pipeline, first 2 steps will run successfully. Step 3 deploy will give you an error. For this, you need to Create one “imagedefinitions.json” file and push it to the code commit. The pipeline will run again and you will again get an error on step 3

Roles attach policy

Add permissions to AWSCodePipelineServiceRole-us-west-1-ecs-pipeline

Attach Permissions

[Create policy](#)

[Filter policies](#) ▾

Showing 7 results

Policy name	Type	Used as
AmazonDMSRedshiftS3Role	AWS managed	None
<input checked="" type="checkbox"/> AmazonS3FullAccess	AWS managed	None
AmazonS3ObjectLambdaExecutionRolePolicy	AWS managed	None
AmazonS3OutpostsFullAccess	AWS managed	None
AmazonS3OutpostsReadOnlyAccess	AWS managed	None
AmazonS3ReadOnlyAccess	AWS managed	None
QuickSightAccessForS3StorageManagementAnalyticsReadOnly	AWS managed	None

[Cancel](#) [Attach policy](#)

Feedback English (US) ▾

Identity and Access Management (IAM)

- Dashboard
- Access management
 - User groups
 - Users
 - Roles**
 - Policies
 - Identity providers
 - Account settings
- Access reports
 - Access analyzer
 - Archive rules
 - Analyzers
 - Settings
- Credential report
- Organization activity
- Service control policies (SCPs)

Search IAM

Roles > AWSCodePipelineServiceRole-us-west-1-ecs-pipeline

Summary

Role ARN: arn:aws:iam::008889205193:role/service-role/AWSCodePipelineServiceRole-us-west-1-ecs-pipeline

Role description: Edit

Instance Profile ARNs: [Edit](#)

Path: /service-role/

Creation time: 2021-10-22 20:15 PDT

Last activity: Not accessed in the tracking period

Maximum session duration: 1 hour [Edit](#)

[Permissions](#) [Trust relationships](#) [Tags](#) [Access Advisor](#) [Revoke sessions](#)

Permissions policies (2 policies applied)

[Attach policies](#) [Add inline policy](#)

Policy name	Policy type	X
AmazonS3FullAccess	AWS managed policy	X
AWSCodePipelineServiceRole-us-west-1-ecs-pipeline	Managed policy	X

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#) [Cookie preferences](#)

Edit: Deploy

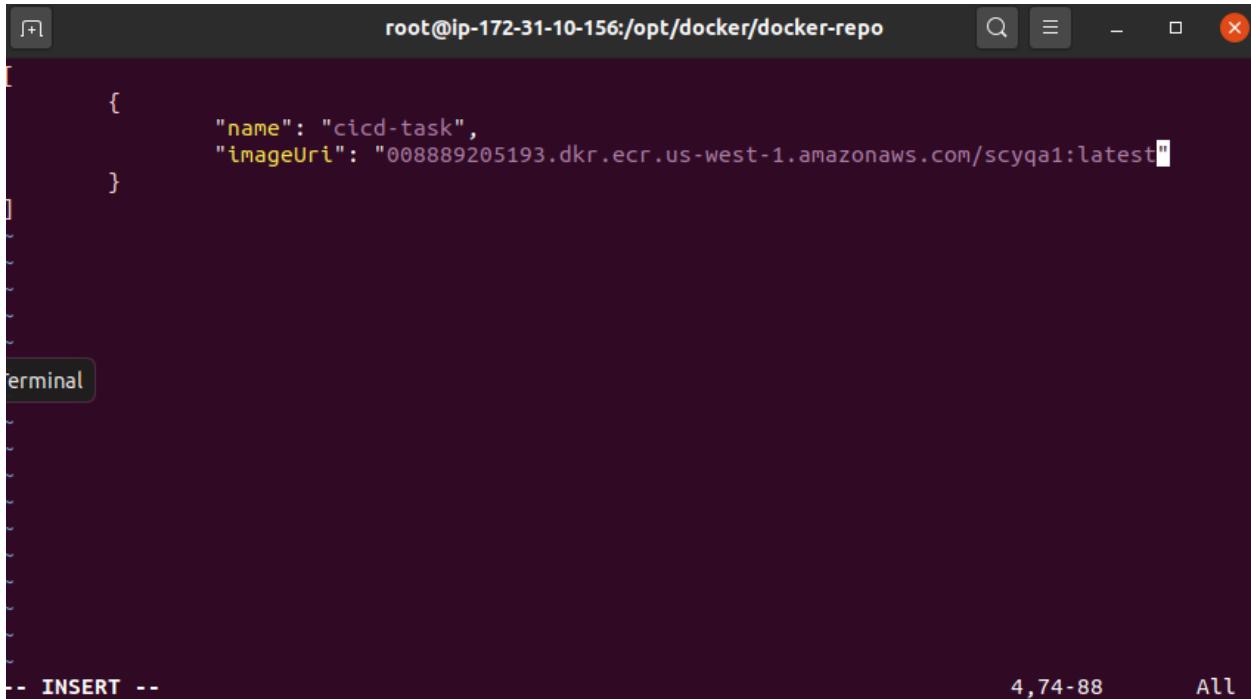
The screenshot shows the AWS CodePipeline 'Edit: Deploy' configuration page. On the left, a sidebar navigation menu includes 'Source' (CodeCommit), 'Build' (CodeBuild), 'Deploy' (CodeDeploy), 'Pipeline' (CodePipeline), 'Getting started', 'Pipelines', 'Pipeline' (selected), 'History', 'Settings', and 'Settings'. Below the sidebar are links for 'Go to resource' and 'Feedback'. The main content area has a header 'Edit: Deploy' with a 'Cancel' button, a 'Delete' button, and a 'Done' button. It features a 'Deploy' action provider set to 'Amazon ECS'. There are buttons for '+ Add action group' and '+ Add action'. At the bottom of the main content area is a '+ Add stage' button. The top of the page includes the AWS logo, a search bar, and account information ('scyqa1', 'N. California', 'Support').

The screenshot shows the 'Edit action' configuration dialog for the 'Deploy' step. The left sidebar lists 'Source', 'Build', 'Deploy' (selected), 'Pipeline', 'Getting started', 'Pipelines', 'Pipeline', 'History', and 'Settings'. The main form fields include:

- Action name:** Deploy
- Action provider:** Amazon ECS
- Region:** US West (N. California)
- Input artifacts:** SourceArtifact
- Cluster name:** cicd-cluster
- Service name:** serviceProject2
- Image definitions file - optional:** imagedefinitions.json
- Deployment timeout - optional:** (empty input field)

At the bottom right of the dialog are 'Cancel', 'Delete', and 'Done' buttons.

Imagedefinitions.json



```
[root@ip-172-31-10-156:/opt/docker/docker-repo]# cat imagedefinitions.json
[{"name": "cicd-task", "imageUri": "008889205193.dkr.ecr.us-west-1.amazonaws.com/scyqa1:latest"}]
```

The terminal window shows the command `cat imagedefinitions.json` being run, displaying its contents. The file contains a single array with one object, defining a task named `cicd-task` with the specified image URI.

Push

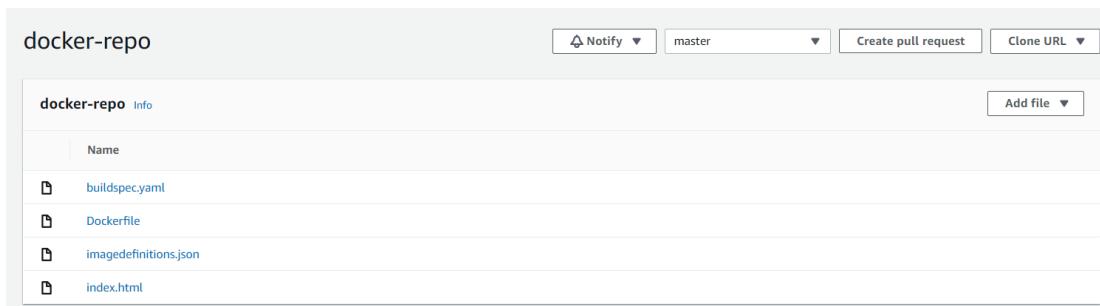
```
[root@ip-172-31-10-156 docker-repo]# git add imagedefinitions.json
[root@ip-172-31-10-156 docker-repo]# git commit -am "imagedefinitions.json"
[master c1c7f0b] imagedefinitions.json
Committer: root <root@ip-172-31-10-156.us-west-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

git config --global --edit

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

1 file changed, 6 insertions(+)
create mode 100644 imagedefinitions.json
[root@ip-172-31-10-156 docker-repo]# git push
```



The GitHub repository page for `docker-repo` shows the following files:

- `buildspec.yaml`
- `Dockerfile`
- `imagedefinitions.json`
- `index.html`

9) Modify Input Artifacts Settings for Stage: Change the input.

Artifacts

The screenshot shows the AWS CodeBuild console with the 'docker-build' project selected. The 'Artifacts' section is highlighted in the configuration sidebar. The build history table shows two successful builds, with the most recent one being a Docker build.

Build run	Status	Build number	Source version	Submitter	Duration	Completed
docker-build:0421113a-83ce-473e-85c9-a519315fde94	Succeeded	9	arn:aws:s3:::codepipeline-us-west-1-971689687851/ecs-pipeline/SourceArti/33hSUJT	codepipeline/ecs-pipeline	1 minute 46 seconds	1 day ago
docker-build:be6c3453-3af2-4a59-bd65-ba301d6d0e1f	Succeeded	8	refs/heads/master	root	1 minute 8 seconds	1 day ago

The screenshot shows the 'Edit Artifacts' dialog for the 'docker-build' project. The 'Artifacts' tab is selected. A primary artifact is configured with the following settings:

- Type:** Amazon S3
- Bucket name:** codepipeline-us-west-1-971689687851
- Name:** output
- Path - optional:** /
- Namespace type - optional:** None
- Artifacts packaging:** None

10) Final Deployment Test and Validation: Push the new version of code in a code commit. It will automatically deploy the new task with the new version. At last, you can run the DNS ALB on your browser

Success

The screenshot shows the AWS CodePipeline console with a successful pipeline execution. The pipeline is named "ecs-pipeline". The execution details show two stages: "Source" and "Build". Both stages are marked as "Succeeded". The "Source" stage was completed 7 minutes ago, and the "Build" stage was completed 5 minutes ago. Both stages used the "AWS CodeCommit" provider and the "d394164f" commit. A "Disable transition" button is visible between the stages. The pipeline status bar indicates "Release change" is available.

The screenshot shows a web browser window with the URL "dockerproject2-1404174096.us-west-1.elb.amazonaws.com". The page content consists of repeated text: "I am neu Cloud student I am neu Cloud student I am neu Cloud student I am neu Cloud student". This indicates that the new code commit has been successfully deployed and is running in the application.

PART II: Design Character Recognition System using AWS Lambda Function [100 points]

- 1) Docker Read the data from S3 (It is OK if you don't/can't use docker but use S3 to store your character image file)

Assigned role with S3 read/write permission and Amazon Rekognition API

User name* test-recognition
 Add another user

Select AWS access type
Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access key - Programmatic access
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

Password - AWS Management Console access
Enables a **password** that allows users to sign-in to the AWS Management Console.

* Required

Feedback English (US) ▾ Services ▾ Search for services, features, marketplace products, and docs [Alt+S] scyqa1 Global ▾ Support ▾ 1 2 3 4 5 Cancel Next: Permissions © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Set permissions

Add user to group Copy permissions from existing user Attach existing policies directly

Create policy

Filter policies ▾ Q: rekog Showing 4 results

Policy name ▾	Type	Used as
<input type="checkbox"/> AmazonRekognitionCustomLabelsFullAccess	AWS managed	None
<input checked="" type="checkbox"/> AmazonRekognitionFullAccess	AWS managed	None
<input type="checkbox"/> AmazonRekognitionReadOnlyAccess	AWS managed	None
<input type="checkbox"/> AmazonRekognitionServiceRole	AWS managed	None

Set permissions boundary

Cancel Previous Next: Tags Feedback English (US) ▾ © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

AWS Services ▾ Search for services, features, marketplace products, and docs [Alt+S] scyqa1 Global Support

1 2 3 4 5

Add user

Set permissions

Add user to group Copy permissions from existing user Attach existing policies directly Create policy Filter policies Q s3 Showing 7 results

Policy name	Type	Used as
AmazonDMSRedshiftS3Role	AWS managed	None
<input checked="" type="checkbox"/> AmazonS3FullAccess	AWS managed	Permissions policy (1)
AmazonS3ObjectLambdaExecutionRolePolicy	AWS managed	None
AmazonS3OutpostsFullAccess	AWS managed	None
AmazonS3OutpostsReadOnlyAccess	AWS managed	None
AmazonS3ReadOnlyAccess	AWS managed	None
QuickSightAccessForS3StorageManagementAnalyticsReadOnly	AWS managed	None

Set permissions boundary

Cancel Previous Next: Tags

Feedback English (US) © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

AWS Services ▾ Search for services, features, marketplace products, and docs [Alt+S] scyqa1 Global Support

1 2 3 4 5

Add user

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	test-recognition
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	AmazonRekognitionFullAccess
Managed policy	AmazonS3FullAccess

Tags

No tags were added.

Cancel Previous Create user

Feedback English (US) © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Configure



```
credentials - Notepad
File Edit Format View Help
[default]
aws_access_key_id = AKIAQEEOWQHEXQGR5C5C
aws_secret_access_key = 73FGkbckJBFBvKAWg4hqxAmh1vMv08Ze7rwfskj|
```



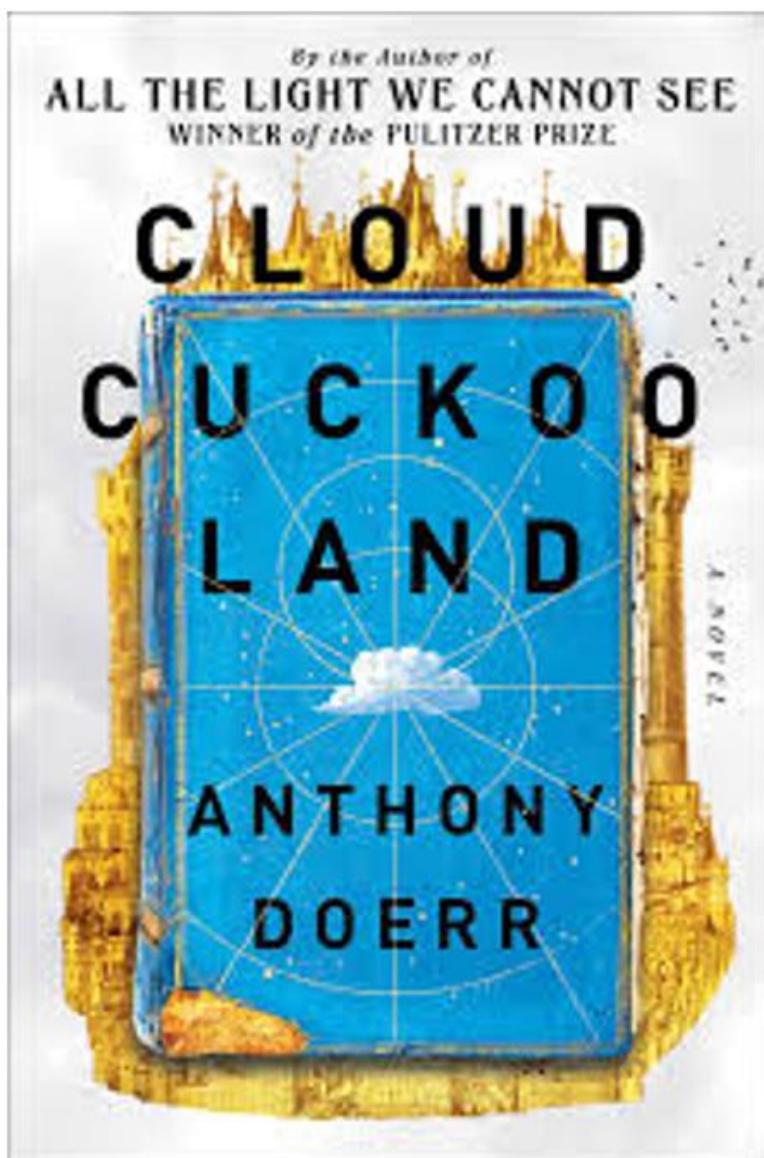
```
config - Notepad
File Edit Format View Help
[default]
region = us-west-1|
```

▶ This PC > Local Disk (C:) > Users > Administrator > .aws

Name	Date modified
config	2021/10/10 10:40:00
credentials	2021/10/10 10:40:00

Upload the image that contains text to your S3 bucket.

Used Image:



Upload:

```
[root@ip-172-31-10-156 s3]# aws s3 cp ./ s3://project2-image/ --recursive
upload: ./imageFile.jpg to s3://project2-image/imageFile.jpg
```

Result:

The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with 'Services' dropdown, search bar, and user info ('scyqa1'). Below it, a blue banner says '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.' The main area shows the 'project2-image' bucket. On the left, there's a sidebar with 'Objects (1)'. The main content area displays a table with one row:

Name	Type	Last modified	Size	Storage class
imageFile.jpg	jpg	October 24, 2021, 16:04:29 (UTC-07:00)	18.9 KB	Standard

At the bottom, there are links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

2) Recognize the character present in the text image you placed in the S3 bucket, using AWS API

For this question, I used two methods.

First method: console

Create lambda function

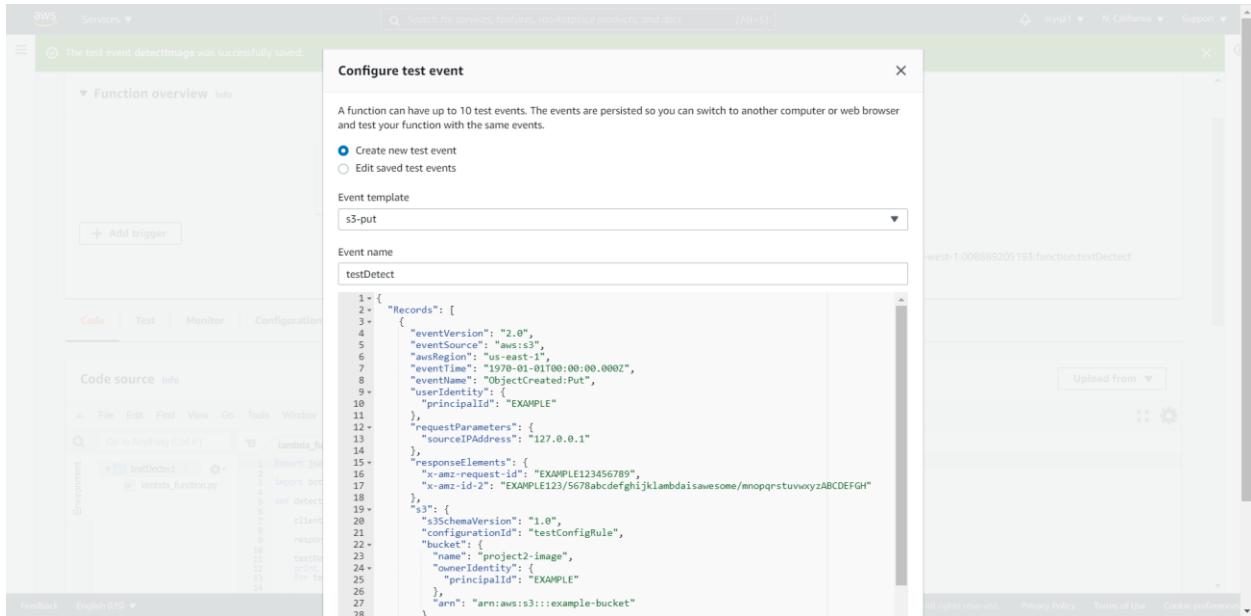
The screenshot shows the 'Create function' wizard in the AWS Lambda console. The 'Blueprints' step is selected, indicated by a blue border around the 'Use a blueprint' section. This section contains a search bar with the query 's3*', a 'Clear filters' button, and a list of three matches:

- s3-get-object-python**: An Amazon S3 trigger that retrieves metadata for the object that has been updated. Trigger type: Python 3.7 - s3.
- rekognition-python**: An Amazon S3 trigger that uses rekognition APIs to detect faces. Trigger type: Python - rekognition - s3.
- s3-get-object**: An Amazon S3 trigger that retrieves metadata for the object that has been updated. Trigger type: Node.js - s3.

At the bottom right of the wizard are 'Cancel' and 'Configure' buttons.

The screenshot shows the 'Configure blueprint s3-get-object-python' step. Under 'Basic information', the function name is set to 'detectimages'. The 'Execution role' section shows 'Use an existing role' selected, with 'service-role/textDetect-role-dz24fcpv' chosen from a dropdown. Under 'S3 trigger', the 'Bucket' is set to 'project2-image' and the 'Event type' is set to 'All object create events'.

Configure test event



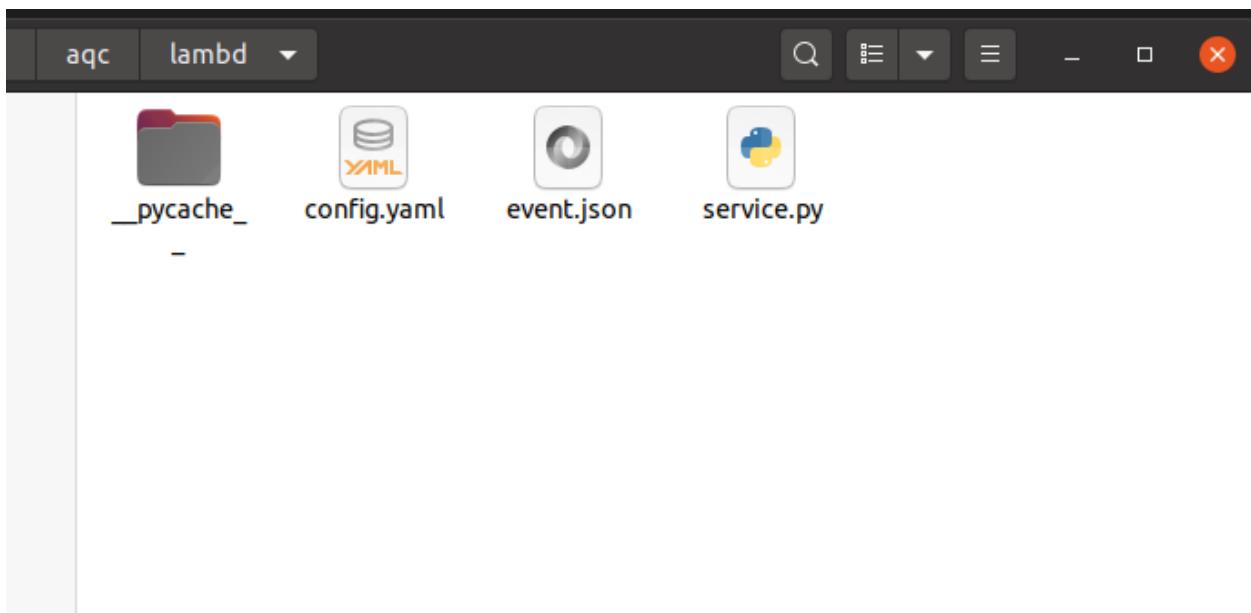
Second method: coding

Install lambda

```
root@ubuntu:/home/aqc/aqc/lambd# pip install python-lambda
Collecting python-lambda
  Downloading python_lambda-11.8.0-py3-none-any.whl (17 kB)
Collecting boto3>=1.4.4
  Downloading boto3-1.19.2-py3-none-any.whl (131 kB)
|██████████| 131 kB 1.9 MB/s
C Rhythmbox click==6.6
  Downloading click-6.6-py2.py3-none-any.whl (71 kB)
|██████████| 71 kB 3.8 MB/s
Collecting PyYAML==5.1
  Downloading PyYAML-5.1.tar.gz (274 kB)
|██████████| 274 kB 3.9 MB/s
Collecting botocore<1.23.0,>=1.22.2
  Downloading botocore-1.22.2-py3-none-any.whl (8.0 MB)
|██████████| 8.0 MB 3.3 MB/s
Collecting jmespath<1.0.0,>=0.7.1
  Downloading jmespath-0.10.0-py2.py3-none-any.whl (24 kB)
Collecting s3transfer<0.6.0,>=0.5.0
  Downloading s3transfer-0.5.0-py3-none-any.whl (79 kB)
```

Initialize the project

```
# 0.10.0 python lambd 11.0.0 s3transfer 0.3.0
root@ubuntu:/home/aqc/aqc/lambd# lambda init
```



Config.yaml

```
[+]
root@ubuntu: /home/aqc/aqc/lambd
region: us-west-1

function_name: my_lambda_function
handler: service.handler
description: My first lambda function
runtime: python3.8
# role: lambda_basic_execution

# S3 upload requires appropriate role with s3:PutObject permission
# (ex. basic_s3_upload), a destination bucket, and the key prefix
# bucket_name: 'example-bucket'
# s3_key_prefix: 'path/to/file/'

# if access key and secret are left blank, boto will use the credentials
# defined in the [default] section of ~/.aws/credentials
aws_access_key_id:AKIAQEEOWQHEXQGR5C5C
aws_secret_access_key:AKIAQEEOWQHEXQGR5C5C

# dist_directory: dist
# timeout: 15
# memory_size: 512
# concurrency: 500
#
```

Service.py

```
root@ubuntu:/home/aqc/aqc/lambd
import json
import urllib.parse
import boto3

print('Loading function')

s3 = boto3.client('s3')

def detect_text(photo, bucket):

    client=boto3.client('rekognition')

    response=client.detect_text(Image={'S3Object':{'Bucket':project2-image,'Name':imageFile.jpg}})

    textDetections=response['TextDetections']
    print ('Detected text\n-----')
    for text in textDetections:
        print ('Detected text:' + text['DetectedText'])
        print ('Confidence: ' + "{:.2f}".format(text['Confidence']) + "%")
        print ('Id: {}'.format(text['Id']))
        if 'ParentId' in text:
            print ('Parent Id: {}'.format(text['ParentId']))
19,1           11%
```

Running result

```
root@ubuntu:/home/aqc/aqc/lambd# lambda invoke -v
Loading function
Detected text
-----
Detected text:Be the Auther of
Confidence: 65.88%
Id: 0
Type:LINE

Detected text:ALL THE LIGHT WE CANNOT SEE
Confidence: 100.00%
Id: 1
Type:LINE

Detected text:WINNER of the Pulitzer PRIZE
Confidence: 94.12%
Id: 2
Type:LINE

Detected text:CLOUD
Confidence: 94.89%
Id: 3
Type:LINE
```

```
Detected text:2340K
Confidence: 41.94%
Id: 27
Parent Id: 6
Type:WORD

Detected text:ANTHONY
Confidence: 100.00%
Id: 28
Parent Id: 7
Type:WORD

Detected text:DOERR
Confidence: 100.00%
Id: 29
Parent Id: 8
Type:WORD

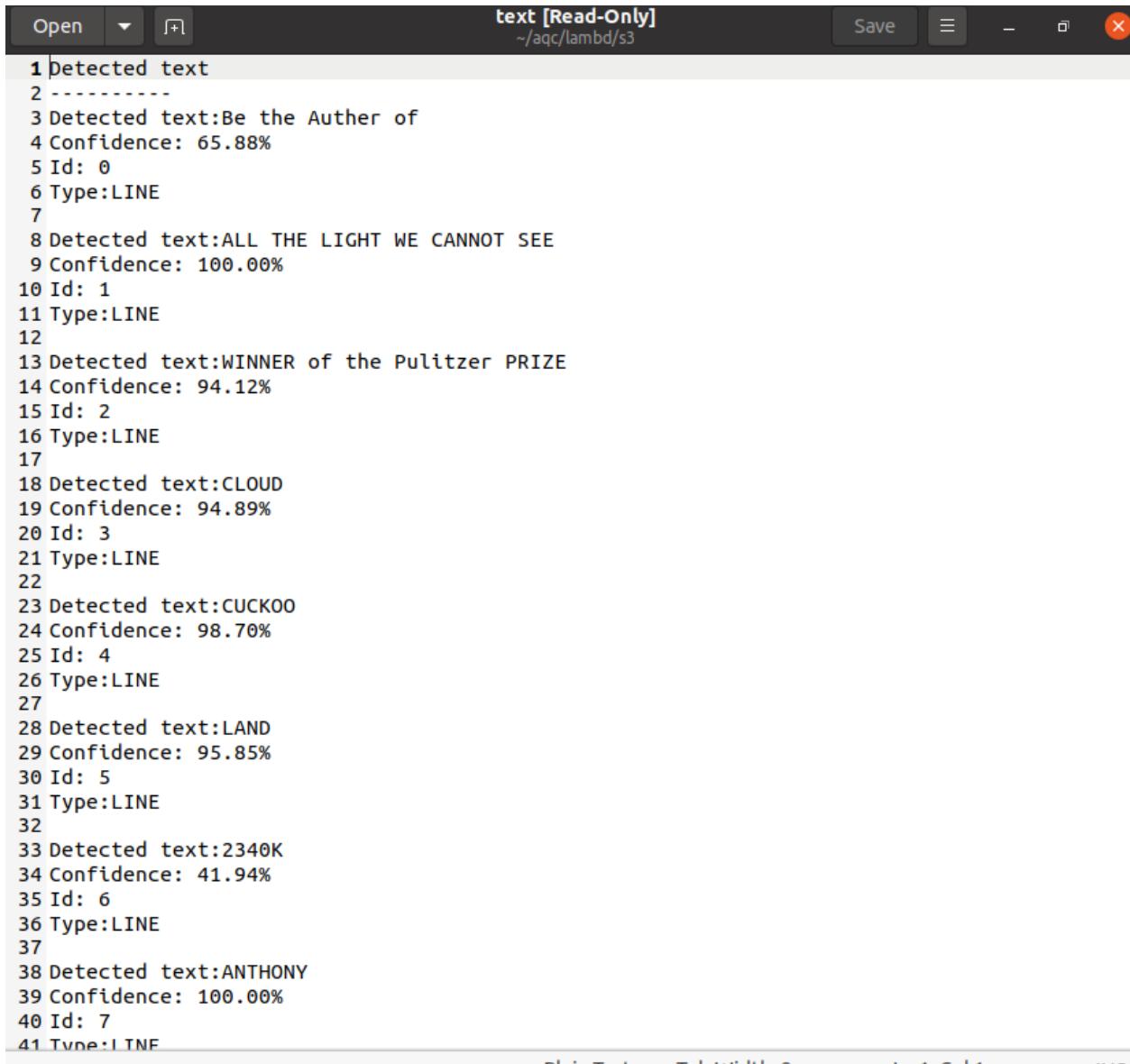
Text detected: 30
None

execution time: 1.25934410s
function execution timeout: 15s
root@ubuntu:/home/aqc/aqc/lambd#
```

Deploy to website

```
root@ubuntu:/home/aqc/aqc/lambd# lambda deploy
Gathering pip packages
```

3) Upload the extracted text from the above step back to S3 bucket. This is your Output (Result) text file.



The screenshot shows a text editor window titled "text [Read-Only]" with the path "~/aqc/lambd/s3". The text content is a list of detected text elements, each numbered from 1 to 41. The elements include detected text like "Be the Author of", "ALL THE LIGHT WE CANNOT SEE", "WINNER of the Pulitzer PRIZE", "CLOUD", "CUCKOO", "LAND", "2340K", "ANTHONY", and "TINF". Each element is followed by its confidence percentage (e.g., 65.88%, 100.00%, 94.12%, etc.) and its type (e.g., LINE). The editor has standard toolbar buttons for Open, Save, and Close, and status bar indicators for Plain Text, Tab Width: 8, Ln 1, Col 1, and INS.

```
1 Detected text
2 -----
3 Detected text:Be the Author of
4 Confidence: 65.88%
5 Id: 0
6 Type:LINE
7
8 Detected text:ALL THE LIGHT WE CANNOT SEE
9 Confidence: 100.00%
10 Id: 1
11 Type:LINE
12
13 Detected text:WINNER of the Pulitzer PRIZE
14 Confidence: 94.12%
15 Id: 2
16 Type:LINE
17
18 Detected text:CLOUD
19 Confidence: 94.89%
20 Id: 3
21 Type:LINE
22
23 Detected text:CUCKOO
24 Confidence: 98.70%
25 Id: 4
26 Type:LINE
27
28 Detected text:LAND
29 Confidence: 95.85%
30 Id: 5
31 Type:LINE
32
33 Detected text:2340K
34 Confidence: 41.94%
35 Id: 6
36 Type:LINE
37
38 Detected text:ANTHONY
39 Confidence: 100.00%
40 Id: 7
41 Type:LINE
```

```
[root@ip-172-31-10-156 s3]# aws s3 cp ./static s3://project2-image/static --recursive
upload: static/text to s3://project2-image/static/text
[root@ip-172-31-10-156 s3]#
```

aws Services ▾

Amazon S3

Buckets

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

Access analyzer for S3

Block Public Access settings for this account

▼ Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight 3

▶ AWS Marketplace for S3

Search for services, features, marketplace products, and docs [Alt+S]

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.

Provide feedback X

Amazon S3 > project2-image

project2-image Info

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

C Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix

Name	Type	Last modified	Size	Storage class
imageFile.jpg	jpg	October 24, 2021, 16:04:29 (UTC-07:00)	18.9 KB	Standard
static/	Folder	-	-	-

Feedback English (US) ▾

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

aws Services ▾

Amazon S3

Buckets

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

Access analyzer for S3

Block Public Access settings for this account

▼ Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight 3

▶ AWS Marketplace for S3

Search for services, features, marketplace products, and docs [Alt+S]

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.

Provide feedback X

Amazon S3 > project2-image > static/

static/

Copy S3 URI

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

C Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix

Name	Type	Last modified	Size	Storage class
text	-	October 26, 2021, 03:42:01 (UTC-07:00)	2.0 KB	Standard