PGPCC | Project

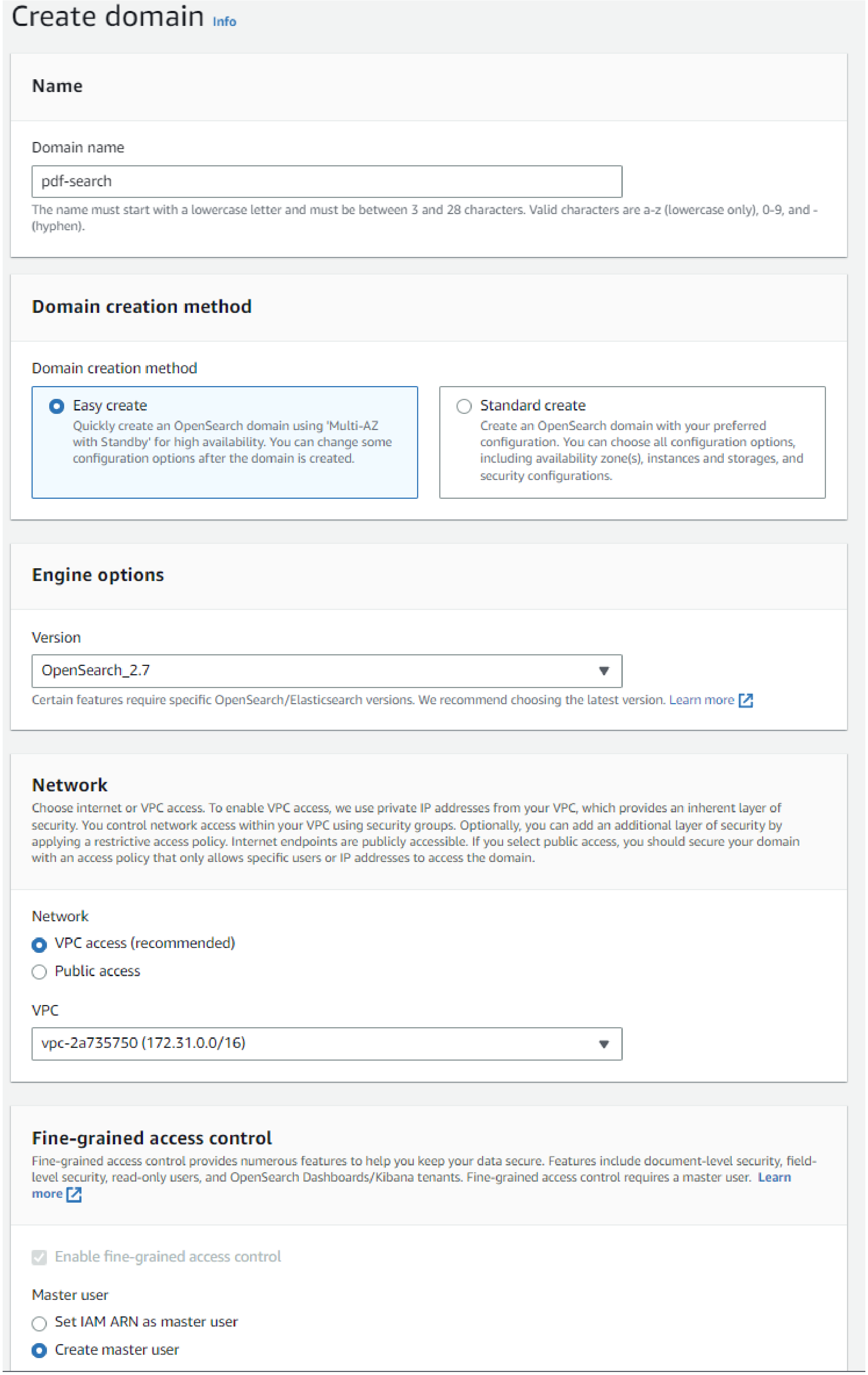
Deploying a search engine using AWS Managed Services

Steps for implementation:

A.) Create an AWS Opensearch domain

Screen shot of creating the domain. The instructions said to use “VPC access” but that led to a lot of problems for a lot of students. So we got the okay to make it public. But I don’t want to take short cuts. I want to do as much as possible to do it the way we would do it on the job. Public ElasticSearch/Opensearch isn’t how we would do it. So I stuck with “VPC Access”

I did everything in the default VPC.



Create domain Info

Name

Domain name

pdf-search

The name must start with a lowercase letter and must be between 3 and 28 characters. Valid characters are a-z (lowercase only), 0-9, and - (hyphen).

Domain creation method

Domain creation method

● Easy create
Quickly create an OpenSearch domain using 'Multi-AZ with Standby' for high availability. You can change some configuration options after the domain is created.

○ Standard create
Create an OpenSearch domain with your preferred configuration. You can choose all configuration options, including availability zone(s), instances and storages, and security configurations.

Engine options

Version

OpenSearch_2.7 ▼

Certain features require specific OpenSearch/Elasticsearch versions. We recommend choosing the latest version. Learn more

Network

Choose internet or VPC access. To enable VPC access, we use private IP addresses from your VPC, which provides an inherent layer of security. You control network access within your VPC using security groups. Optionally, you can add an additional layer of security by applying a restrictive access policy. Internet endpoints are publicly accessible. If you select public access, you should secure your domain with an access policy that only allows specific users or IP addresses to access the domain.

Network

● VPC access (recommended)

○ Public access

VPC

vpc-2a735750 (172.31.0.0/16) ▼

Fine-grained access control

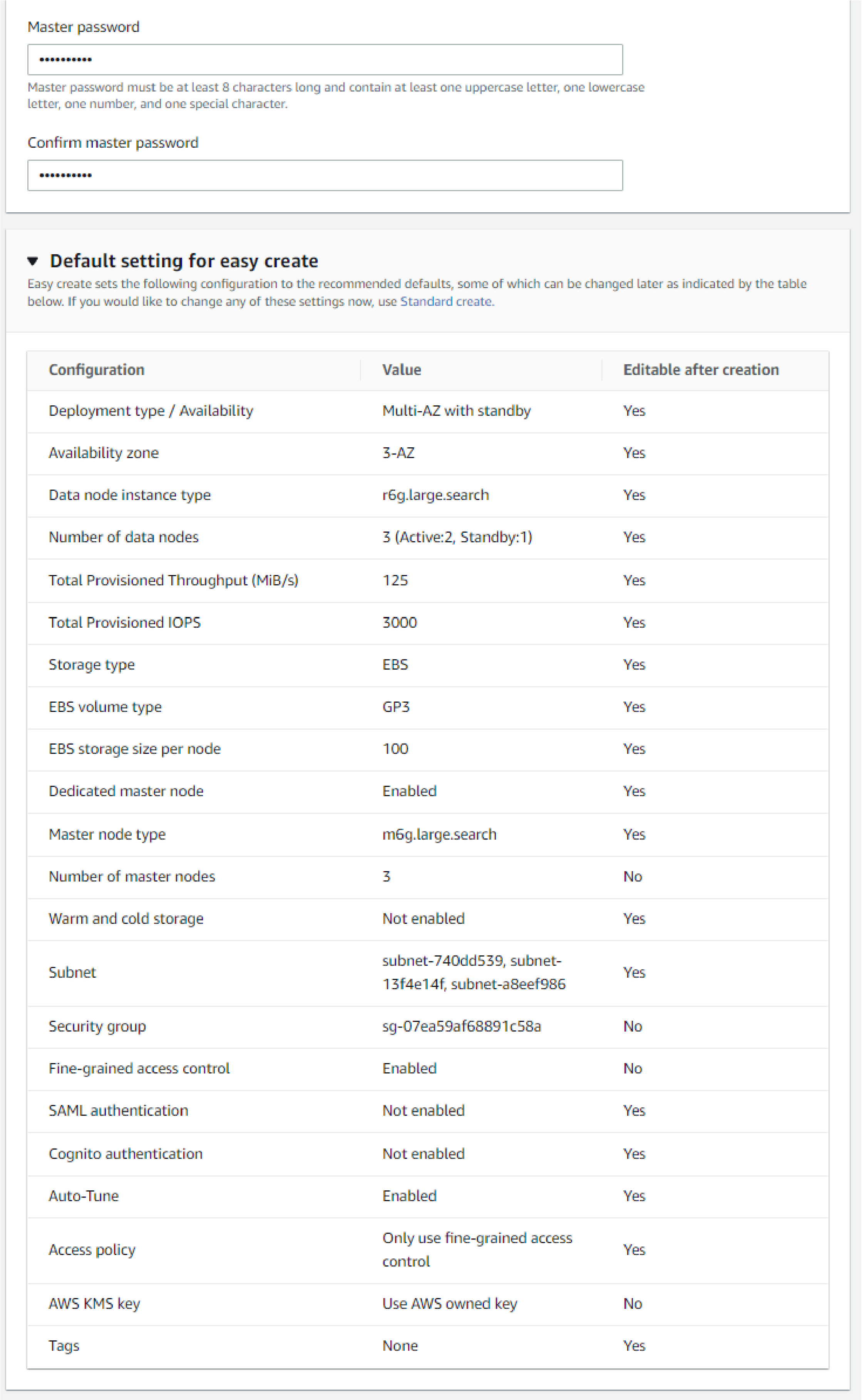
Fine-grained access control provides numerous features to help you keep your data secure. Features include document-level security, field-level security, read-only users, and OpenSearch Dashboards/Kibana tenants. Fine-grained access control requires a master user. Learn more

☑ Enable fine-grained access control

Master user

○ Set IAM ARN as master user

● Create master user



Master password

••••••••••

Master password must be at least 8 characters long and contain at least one uppercase letter, one lowercase letter, one number, and one special character.

Confirm master password

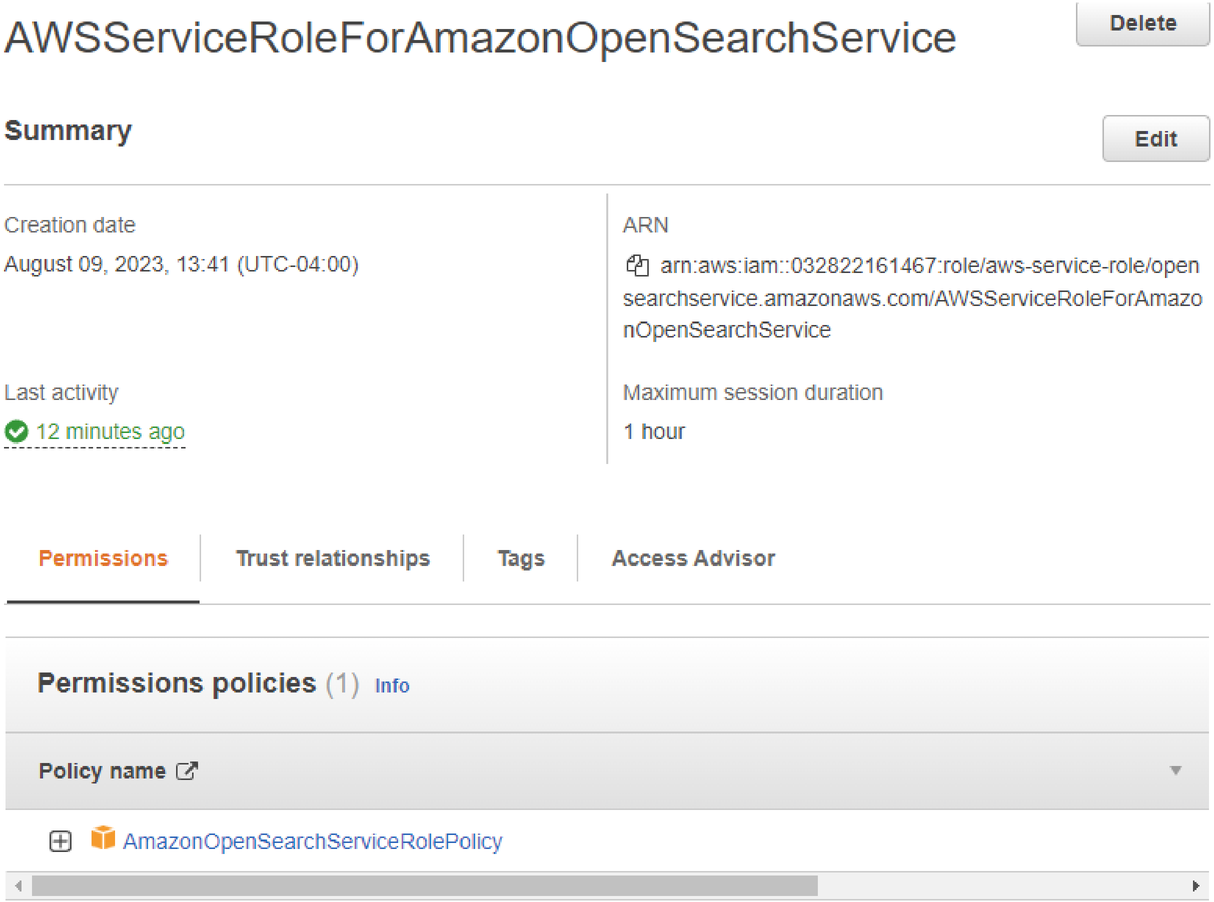
••••••••••

▼ Default setting for easy create

Easy create sets the following configuration to the recommended defaults, some of which can be changed later as indicated by the table below. If you would like to change any of these settings now, use Standard create.

Configuration	Value	Editable after creation
Deployment type / Availability	Multi-AZ with standby	Yes
Availability zone	3-AZ	Yes
Data node instance type	r6g.large.search	Yes
Number of data nodes	3 (Active:2, Standby:1)	Yes
Total Provisioned Throughput (MiB/s)	125	Yes
Total Provisioned IOPS	3000	Yes
Storage type	EBS	Yes
EBS volume type	GP3	Yes
EBS storage size per node	100	Yes
Dedicated master node	Enabled	Yes
Master node type	m6g.large.search	Yes
Number of master nodes	3	No
Warm and cold storage	Not enabled	Yes
Subnet	subnet-740dd539, subnet-13f4e14f, subnet-a8eef986	Yes
Security group	sg-07ea59af68891c58a	No
Fine-grained access control	Enabled	No
SAML authentication	Not enabled	Yes
Cognito authentication	Not enabled	Yes
Auto-Tune	Enabled	Yes
Access policy	Only use fine-grained access control	Yes
AWS KMS key	Use AWS owned key	No
Tags	None	Yes

This IAM role was created by the previous step



AWSServiceRoleForAmazonOpenSearchService

Summary

Creation date
August 09, 2023, 13:41 (UTC-04:00)

ARN
arn:aws:iam::032822161467:role/aws-service-role/opensearchservice.amazonaws.com/AWSServiceRoleForAmazonOpenSearchService

Last activity
✓ 12 minutes ago

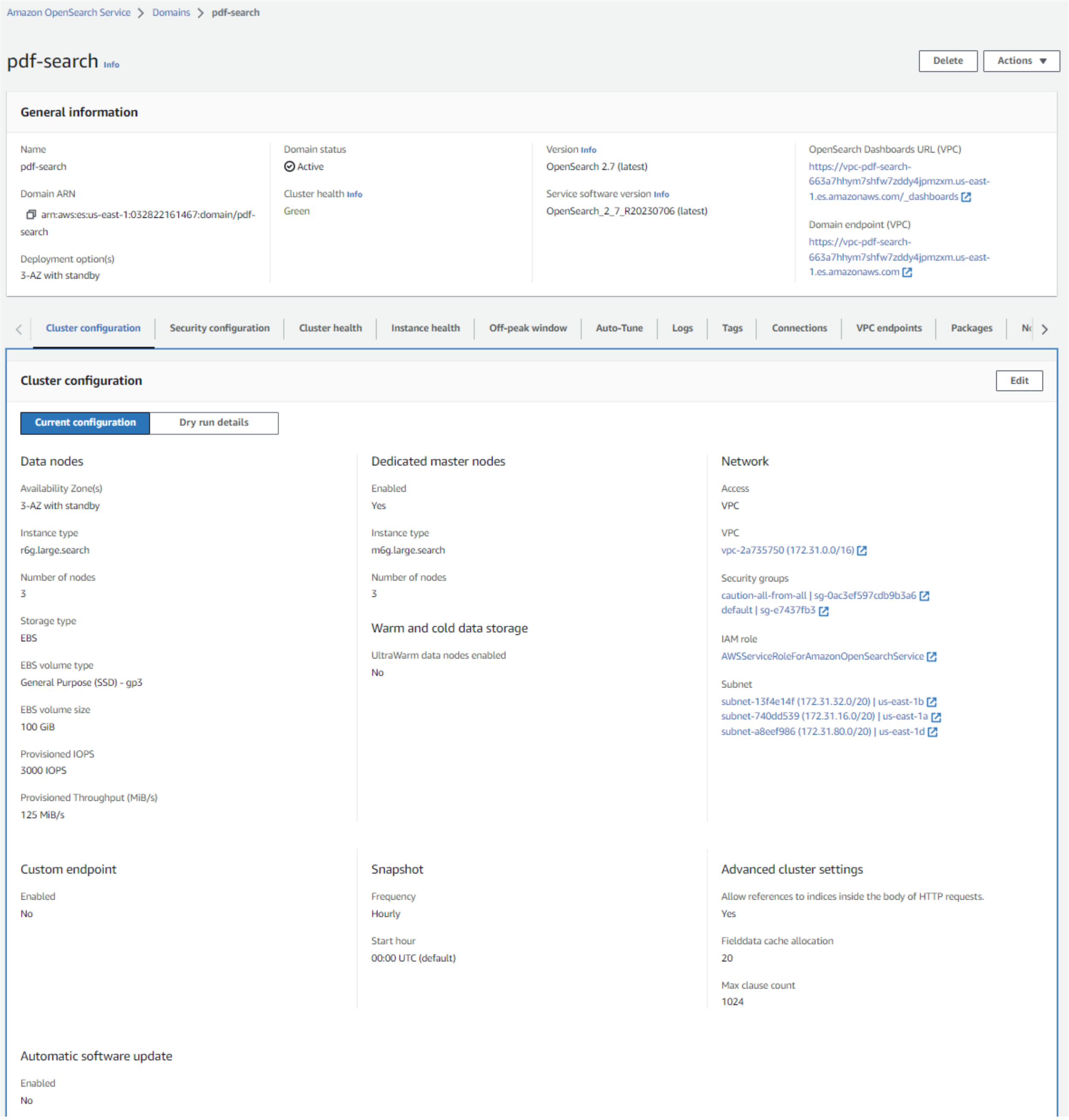
Maximum session duration
1 hour

Permissions policies (1) Info

Policy name
AmazonOpenSearchServiceRolePolicy

The Create Domain steps seemed to randomly pick the subnets to use and the security groups. It did not use the VPC default security group as one might expect. It grabbed my tomcat security group. But I changed it to the default VPC security group. I was struggling getting this to work in a VPC so for testing purposes I also included a security group that opened up all ports for everyone. Obviously before going into production this would have to be removed.

I took note of what subnets were used to make sure all lambda functions were on these subnets.



pdf-search Info

General information

Name
pdf-search

Domain ARN
arn:aws:es:us-east-1:032822161467:domain/pdf-search

Deployment option(s)
3-AZ with standby

Domain status
Active

Cluster health Info
Green

Version Info
OpenSearch 2.7 (latest)

Service software version Info
OpenSearch_2_7_R20230706 (latest)

OpenSearch Dashboards URL (VPC)
https://vpc-pdf-search-663a7hhym7shfw7zddy4jpmzxm.us-east-1.es.amazonaws.com/_dashboards

Domain endpoint (VPC)
https://vpc-pdf-search-663a7hhym7shfw7zddy4jpmzxm.us-east-1.es.amazonaws.com

Cluster configuration

Data nodes

Availability Zone(s)
3-AZ with standby

Instance type
r6g.large.search

Number of nodes
3

Storage type
EBS

EBS volume type
General Purpose (SSD) - gp3

EBS volume size
100 GiB

Provisioned IOPS
3000 IOPS

Provisioned Throughput (MiB/s)
125 MiB/s

Dedicated master nodes

Enabled
Yes

Instance type
m6g.large.search

Number of nodes
3

Warm and cold data storage

UltraWarm data nodes enabled
No

Network

Access
VPC

VPC
vpc-2a735750 (172.31.0.0/16)

Security groups
caution-all-from-all | sg-0ac3ef597cdb9b3a6
default | sg-e7437fb3

IAM role
AWSServiceRoleForAmazonOpenSearchService

Subnet
subnet-13f4e14f (172.31.32.0/20) | us-east-1b
subnet-740dd539 (172.31.16.0/20) | us-east-1a
subnet-a8eef986 (172.31.80.0/20) | us-east-1d

Custom endpoint

Enabled
No

Snapshot

Frequency
Hourly

Start hour
00:00 UTC (default)

Advanced cluster settings

Allow references to indices inside the body of HTTP requests.
Yes

Fielddata cache allocation
20

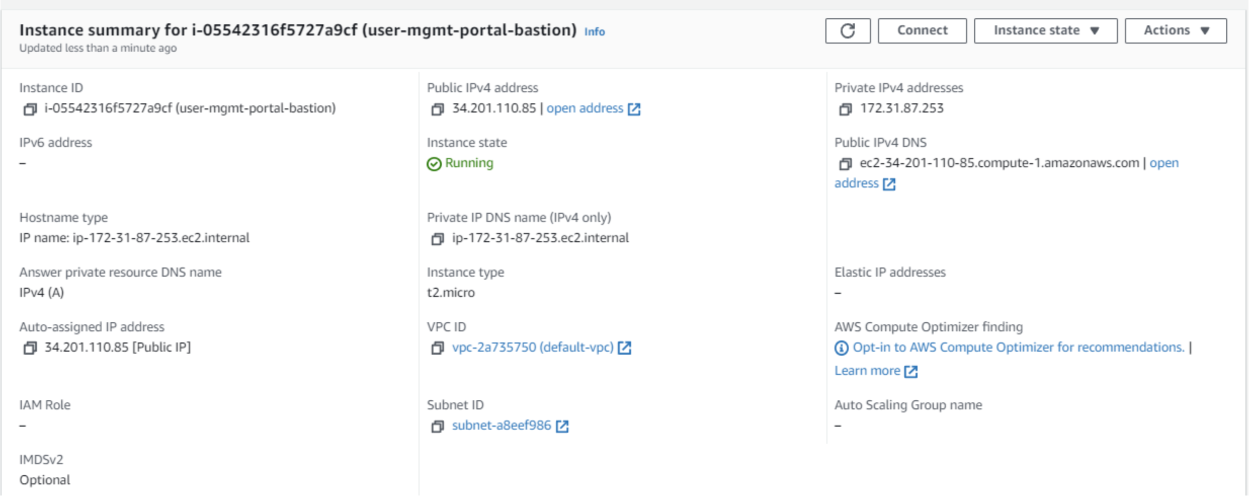
Max clause count
1024

Automatic software update

Enabled
No

The next step is to add a master user.

The necessary configurations were not in place so the open search domain was not accessible from outside the VPC. I setup a SSH tunnel using an existing bastion host I have on my default VPC.



Instance summary for i-05542316f5727a9cf (user-mgmt-portal-bastion) Info

Updated less than a minute ago

Instance ID
i-05542316f5727a9cf (user-mgmt-portal-bastion)

Public IPv4 address
34.201.110.85 | open address

Private IPv4 addresses
172.31.87.253

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-34-201-110-85.compute-1.amazonaws.com | open address

Hostname type
IP name: ip-172-31-87-253.ec2.internal

Private IP DNS name (IPv4 only)
ip-172-31-87-253.ec2.internal

Answer private resource DNS name
IPv4 (A)

Instance type
t2.micro

Elastic IP addresses
–

Auto-assigned IP address
34.201.110.85 [Public IP]

VPC ID
vpc-2a735750 (default-vpc)

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

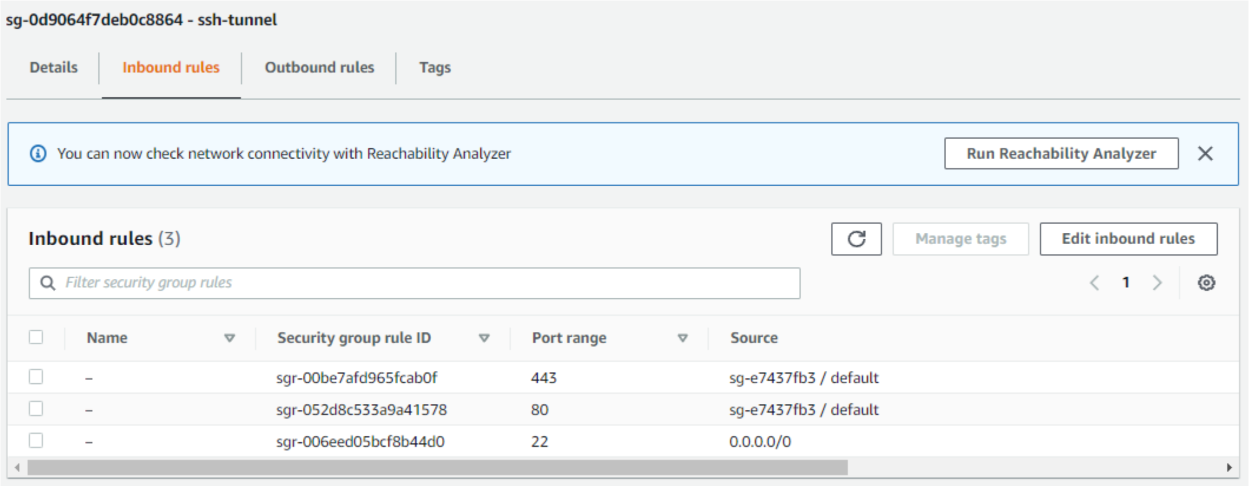
IAM Role
–

Subnet ID
subnet-a8eef986

Auto Scaling Group name
–

IMDSv2
Optional

I created a “ssh-tunnel” security group and assigned it to the instance.



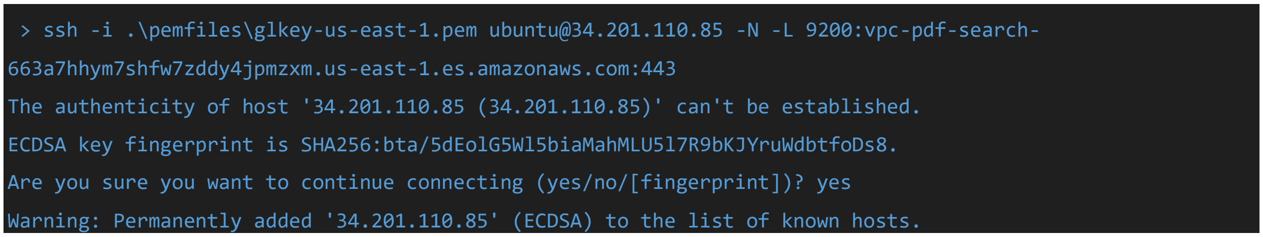
sg-0d9064f7deb0c8864 - ssh-tunnel

You can now check network connectivity with Reachability Analyzer

Inbound rules (3)

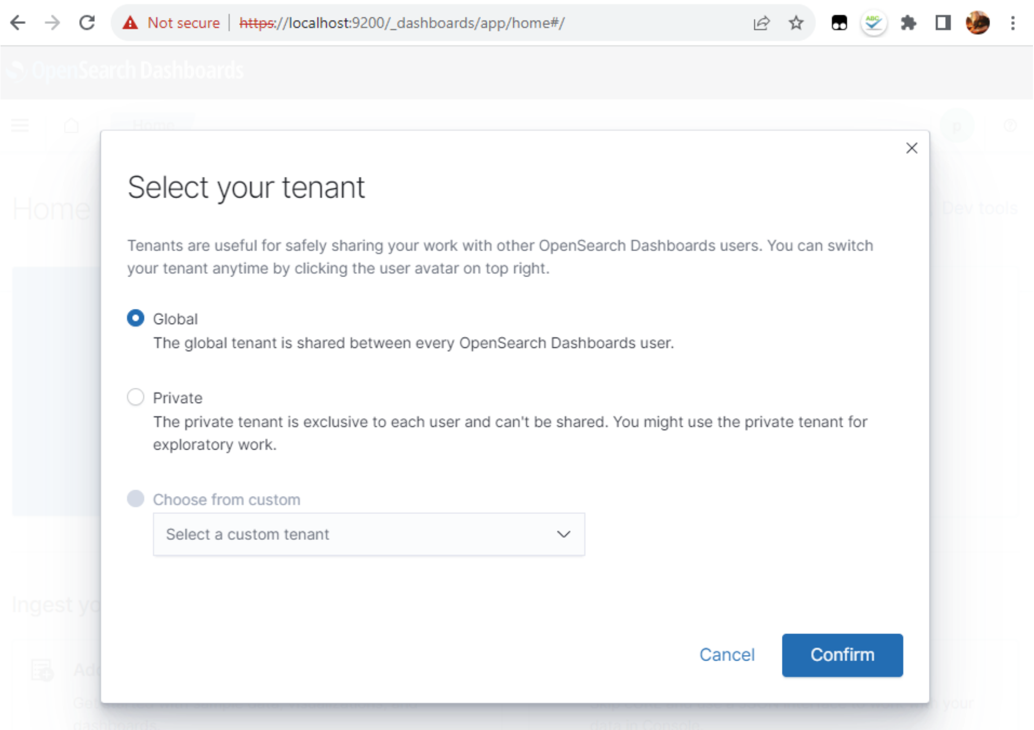
☐	Name	Security group rule ID	Port range	Source
☐	–	sgr-00be7afd965fcab0f	443	sg-e7437fb3 / default
☐	–	sgr-052d8c533a9a41578	80	sg-e7437fb3 / default
☐	–	sgr-006eed05bcf8b44d0	22	0.0.0.0/0

Then I established the SSH tunnel.



```
 > ssh -i .\pemfiles\glkey-us-east-1.pem ubuntu@34.201.110.85 -N -L 9200:vpc-pdf-search-
663a7hhym7shfw7zddy4jpmzxm.us-east-1.es.amazonaws.com:443
The authenticity of host '34.201.110.85 (34.201.110.85)' can't be established.
ECDSA key fingerprint is SHA256:bta/5dEolG5Wl5biaMahMLU5l7R9bKJYruWdbtfoDs8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '34.201.110.85' (ECDSA) to the list of known hosts.
```

I was then able to go to https://localhost:9200 to access the Opensearch dashboard



Not secure | ~~https~~://localhost:9200/_dashboards/app/home#/

Select your tenant

Tenants are useful for safely sharing your work with other OpenSearch Dashboards users. You can switch your tenant anytime by clicking the user avatar on top right.

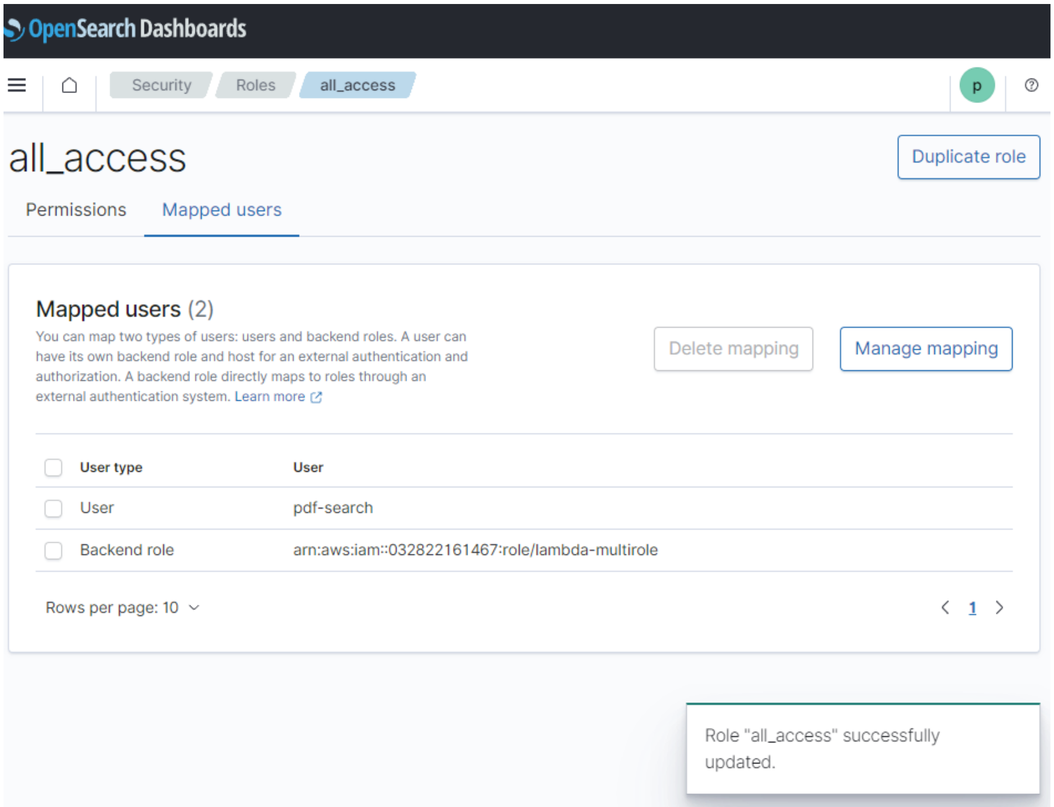
● Global
The global tenant is shared between every OpenSearch Dashboards user.

○ Private
The private tenant is exclusive to each user and can't be shared. You might use the private tenant for exploratory work.

○ Choose from custom
Select a custom tenant

Cancel Confirm

I created the backend role using my lambda-multirole IAM Role. In production I would create a role specify for this task and lock it down with minimal policies needed.



all_access

Mapped users (2)

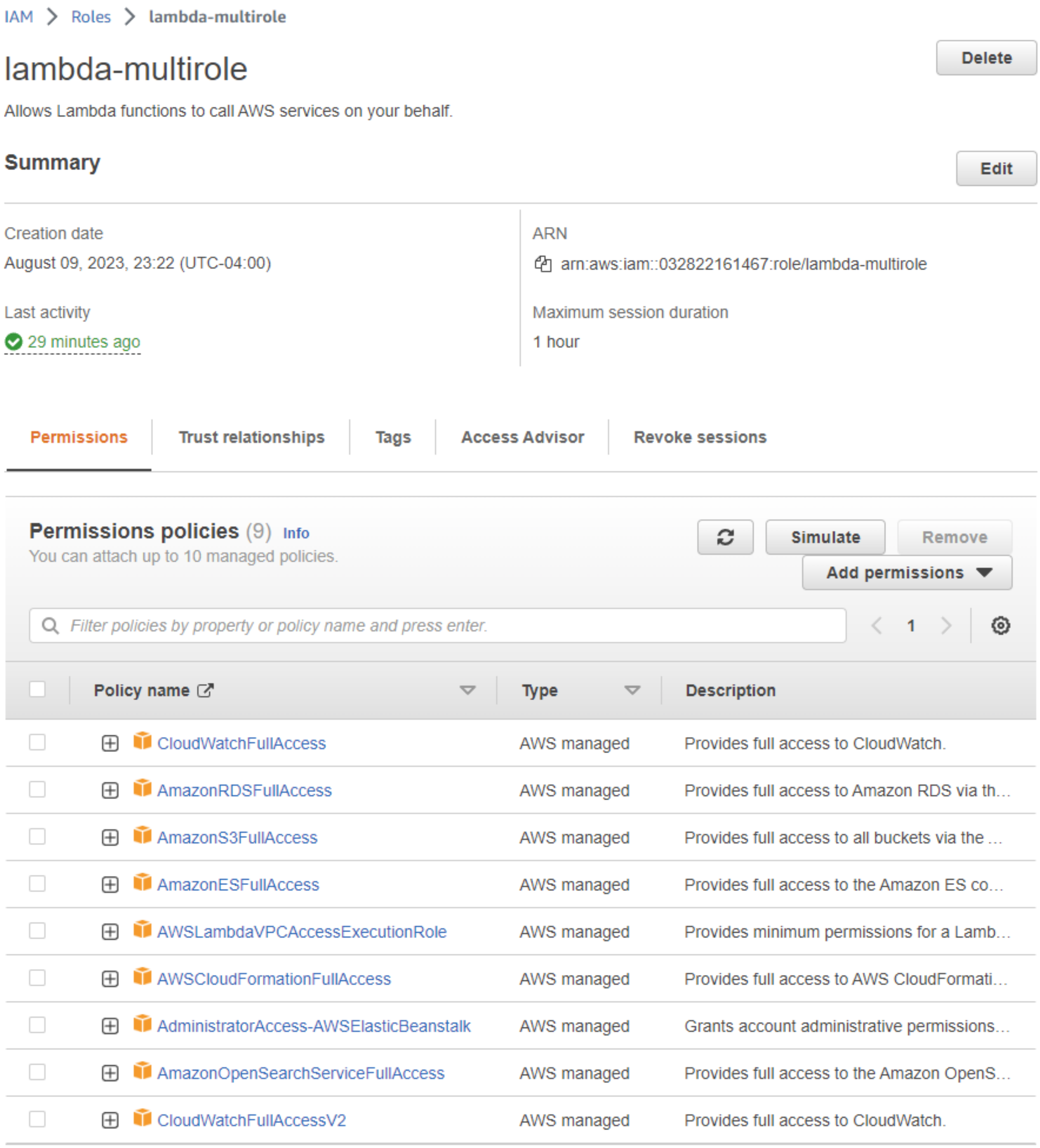
You can map two types of users: users and backend roles. A user can have its own backend role and host for an external authentication and authorization. A backend role directly maps to roles through an external authentication system. Learn more

□ User type	User
□ User	pdf-search
□ Backend role	arn:aws:iam::032822161467:role/lambda-multirole

Rows per page: 10

Role "all_access" successfully updated.

I used lambda-multirole for all lambda IAM Roles. So it has a lot of policies assigned to it. Elastic Beanstalk is assigned for a side project I have running. If this was production I would have dedicated roles with more restrictive policies



lambda-multirole

Allows Lambda functions to call AWS services on your behalf.

Summary

Creation date
August 09, 2023, 23:22 (UTC-04:00)

Last activity
✓ 29 minutes ago

ARN
arn:aws:iam::032822161467:role/lambda-multirole

Maximum session duration
1 hour

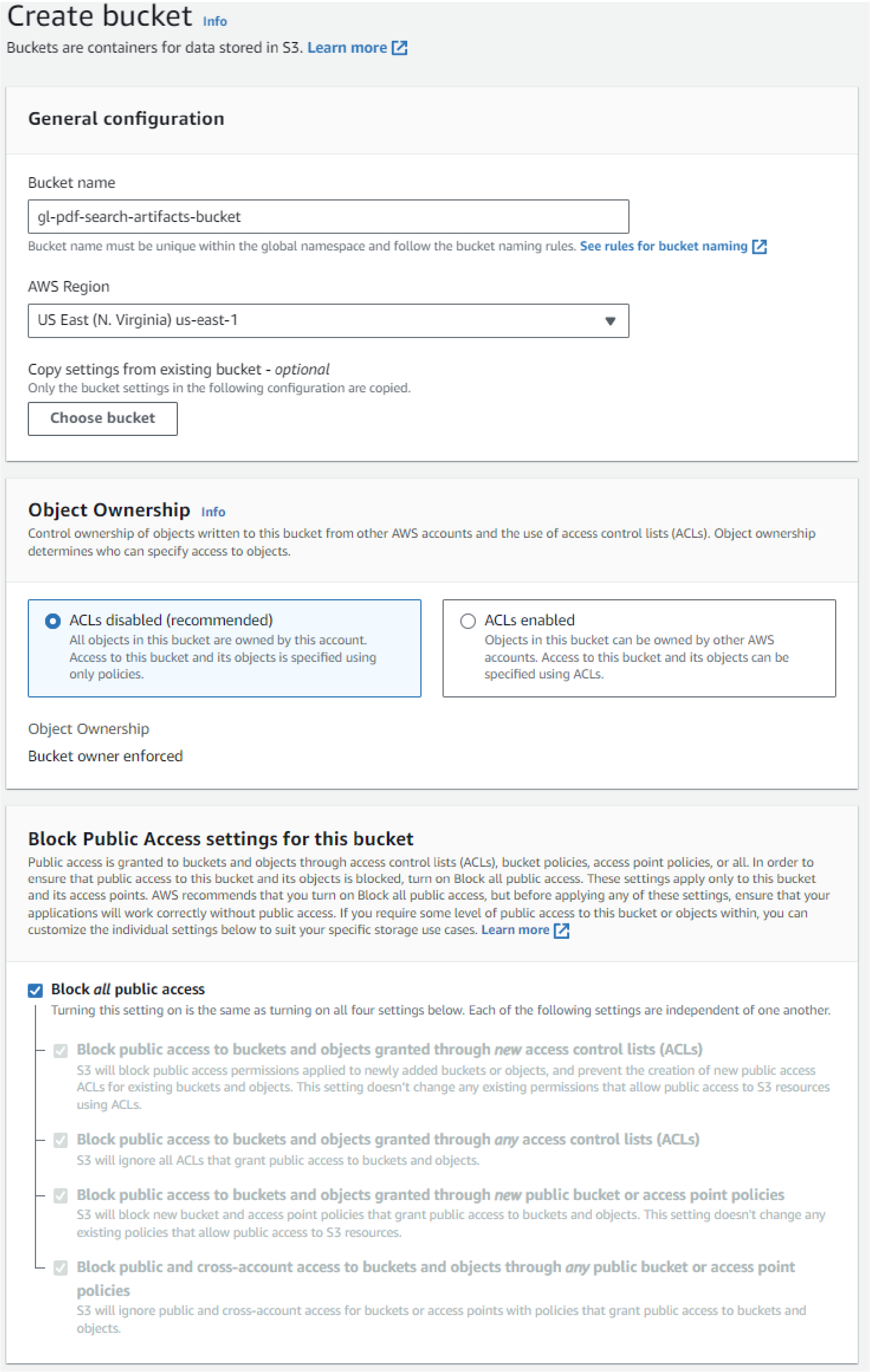
Permissions policies (9)

You can attach up to 10 managed policies.

Policy name	Type	Description
CloudWatchFullAccess	AWS managed	Provides full access to CloudWatch.
AmazonRDSFullAccess	AWS managed	Provides full access to Amazon RDS via th...
AmazonS3FullAccess	AWS managed	Provides full access to all buckets via the ...
AmazonESFullAccess	AWS managed	Provides full access to the Amazon ES co...
AWSLambdaVPCAccessExecutionRole	AWS managed	Provides minimum permissions for a Lamb...
AWSCloudFormationFullAccess	AWS managed	Provides full access to AWS CloudFormati...
AdministratorAccess-AWSElasticBeanstalk	AWS managed	Grants account administrative permissions...
AmazonOpenSearchServiceFullAccess	AWS managed	Provides full access to the Amazon OpenS...
CloudWatchFullAccessV2	AWS managed	Provides full access to CloudWatch.

Steps B, C and D) Download deployment package files, create pipelines to deploy and setup event triggers

The original architecture for this project for each of the lambda functions would call `aws cloudformation package` in the buildspec.yml file for the CodeBuild projects in the CodePipeline. This command packages up all the artifacts needed for cloud formation and stores them in S3 and produces a new template file with the S3 artifact in the templates CodeUri for the CloudFormation Templates. To do this `aws cloudformation package` needs an S3 bucket to store the packages. I created an S3 bucket `gl-pdf-search-artifacts-bucket` for all the buildspec.yml files to use.



Create bucket Info

Buckets are containers for data stored in S3. Learn more

General configuration

Bucket name

gl-pdf-search-artifacts-bucket

Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

AWS Region

US East (N. Virginia) us-east-1 ▼

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

● ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

○ ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership

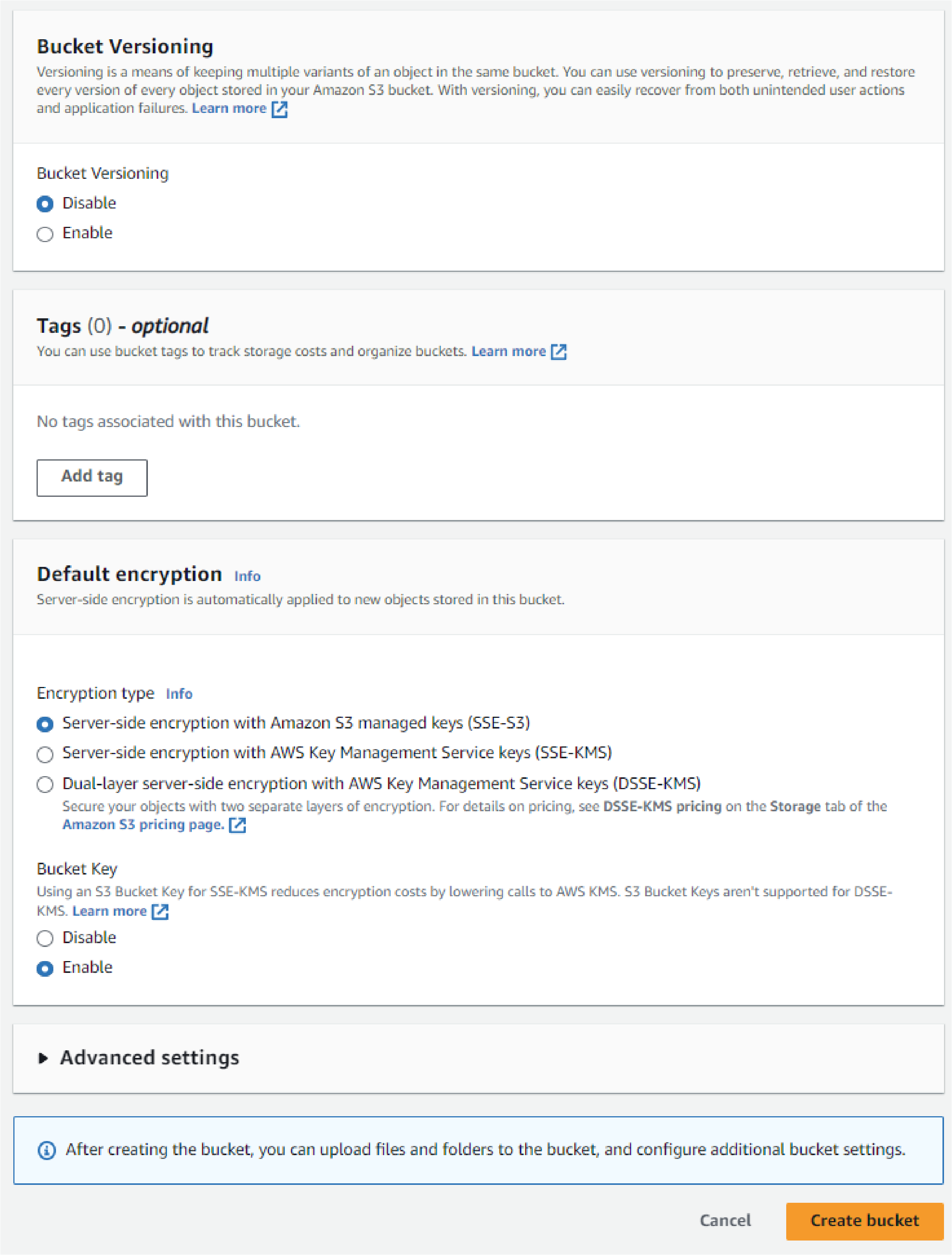
Bucket owner enforced

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

☑ **Block *all* public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- ☑ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
  S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- ☑ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
  S3 will ignore all ACLs that grant public access to buckets and objects.
- ☑ **Block public access to buckets and objects granted through *new* public bucket or access point policies**
  S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- ☑ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
  S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.



Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. **Learn more**

Bucket Versioning

● Disable

○ Enable

Tags (0) - *optional*

You can use bucket tags to track storage costs and organize buckets. **Learn more**

No tags associated with this bucket.

Default encryption Info

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type Info

● Server-side encryption with Amazon S3 managed keys (SSE-S3)

○ Server-side encryption with AWS Key Management Service keys (SSE-KMS)

○ Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)

Secure your objects with two separate layers of encryption. For details on pricing, see **DSSE-KMS pricing** on the **Storage** tab of the **Amazon S3 pricing page.**

Bucket Key

Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. **Learn more**

○ Disable

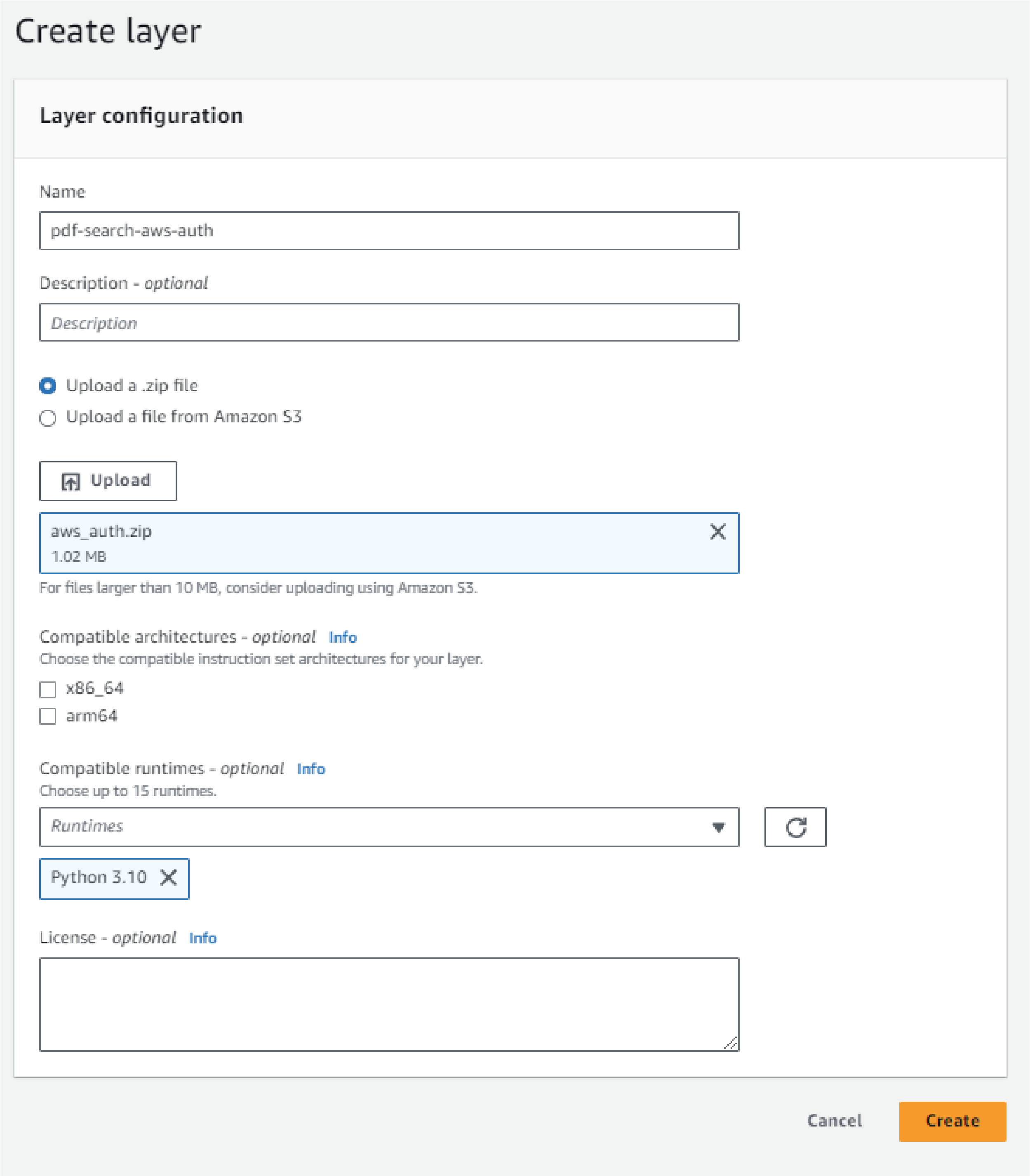
● Enable

▶ Advanced settings

After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

The next step was to create the pipelines. But changes needed to be made to the source files first. Before I could do that, I needed to create Lambda Layers to share dependencies.

pdf-search-aws-auth was the dependencies provided with the project files. I create the following lambda layer with it.



Create layer

Layer configuration

Name

pdf-search-aws-auth

Description - *optional*

Description

● Upload a .zip file

○ Upload a file from Amazon S3

aws_auth.zip
1.02 MB

For files larger than 10 MB, consider uploading using Amazon S3.

Compatible architectures - *optional* Info

Choose the compatible instruction set architectures for your layer.

□ x86_64

□ arm64

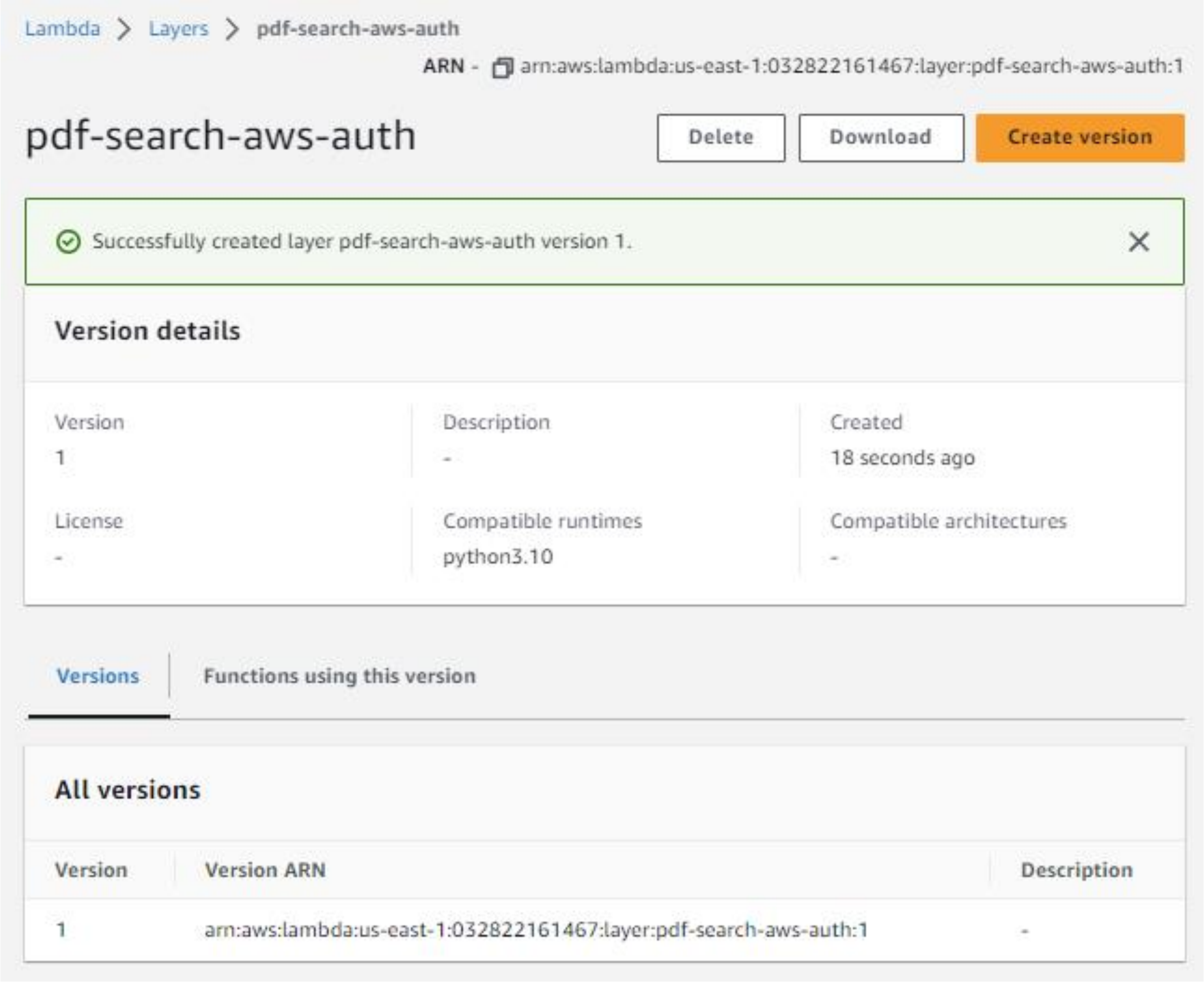
Compatible runtimes - *optional* Info

Choose up to 15 runtimes.

Runtimes

Python 3.10

License - *optional* Info


Lambda > Layers > pdf-search-aws-auth

ARN - arn:aws:lambda:us-east-1:032822161467:layer:pdf-search-aws-auth:1

pdf-search-aws-auth

Delete
Download
Create version

Successfully created layer pdf-search-aws-auth version 1.

Version details

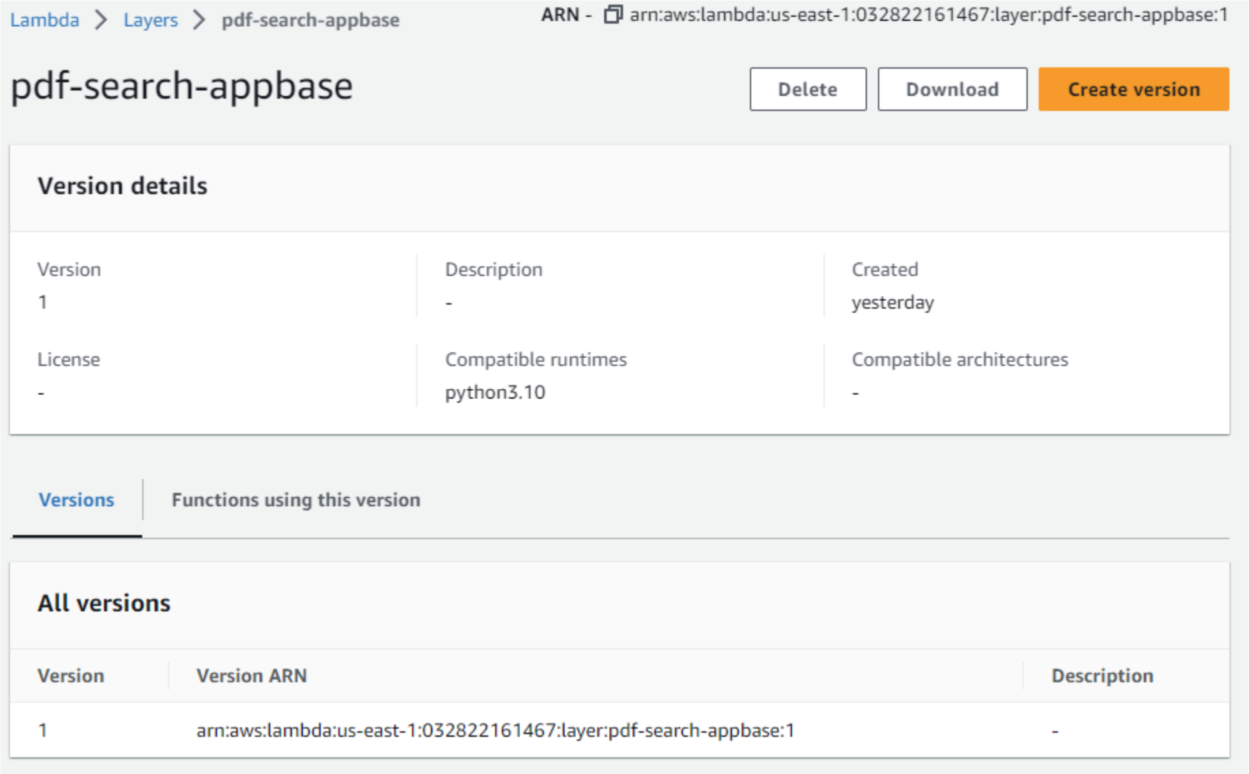
Version	Description	Created
1	-	18 seconds ago
License	**Compatible runtimes**	**Compatible architectures**
-	python3.10	-

Versions
Functions using this version

All versions

Version	Version ARN	Description
1	arn:aws:lambda:us-east-1:032822161467:layer:pdf-search-aws-auth:1	-

The provided project source uses the pypdf module. The dependencies were not included in the source files for the project. But I created a lambda layer for it in a previous project. So I added it as another lambda layer.



Lambda > Layers > pdf-search-appbase

ARN - arn:aws:lambda:us-east-1:032822161467:layer:pdf-search-appbase:1

pdf-search-appbase

Version details

Version 1	Description -	Created yesterday
License -	Compatible runtimes python3.10	Compatible architectures -

All versions

Version	Version ARN	Description
1	arn:aws:lambda:us-east-1:032822161467:layer:pdf-search-appbase:1	-

Then I was able to update the source code and create my pipelines.

PDFtoTXT source files: This lambda function is triggered when PDFs are uploaded to a S3 bucket to extract text from the PDF and store the text in another S3 bucket.

buildspec.yml: This file will be used by the CodeBuild project in the CodePipeline to create a Lambda function to export text from PDFs uploaded to an S3 bucket.

```yaml
version: 0.2
phases:
  install:
    runtime-versions:
      python: 3.10
  build:
    commands:
      - aws cloudformation package --template-file pdftotxt.yaml --s3-bucket gl-pdf-search-artifacts-bucket --output-template-file lambdaoutput.yaml

artifacts:
  type: zip
  files:
    - lambdaoutput.yaml
```

pdftotxt.yaml: My intention was to create a CloudFormation template that would do the following

- Create the PdfToText lambda function
- Create the S3 bucket for documents to be uploaded to
- Create the S3 bucket intermediary storage
- Setup permissions to enable the S3 bucket for the document uploads to invoke the PdfToText lambda function
- Configure a trigger for the S3 bucket to invoke the function whenever a PDF is uploaded.

I wasted too much time on this. I eventually learned the last item on that list could not be done in the cloudformation template without the stack being created at least once without it due to circular dependency problems. I found three solutions to this problem.

1) Create the CloudFormation template without the even trigger and manually add it.
2) Create the CloudFormation stack without it and then update the template to include the event. With this approach the event trigger is added after the stack has deployed the Lambda function and S3 bucket, thus removing the circular dependency.
3) Create a CloudFormation script that would deploy the Lambda function and S3 bucket and then invoke another lambda function after they are created to create the event.

#3 felt like a really bad hack as the example had the code in the template. #1 is prone to error, because it you delete the stack without manually removing the trigger, CloudFormation gets stuck. So I went with option #2. This does present a maintenance problem as it is not intuitive. If someone deletes the stack and attempts to recreate it they will have to know they have to perform this manual step first.

If this was going into production I might look at something like an AWS Event Bridge. I don’t know if it would make sense to go with something like that, but it’s worth looking into a more intuitive solution.

This is the only CloudFormation template that includes both Lambda layers.

```yaml
AWSTemplateFormatVersion: 2010-09-09
Transform: 'AWS::Serverless-2016-10-31'
Description: >-
  Defining the PDF To Text Lambda function and the S3 buckets for the PDF
  documents and the text files
Resources:
  glPdfSearchPdfToTxtLambda:
    Type: 'AWS::Serverless::Function'
    Properties:
      FunctionName: gl-pdf-search-pdf-to-txt-lambda
      Handler: lambda_function.lambda_handler
      Runtime: python3.7
      Description: ''
      CodeUri: .
      MemorySize: 512
      Timeout: 900
      Role: 'arn:aws:iam::032822161467:role/lambda-multirole'
      Environment:
        Variables:
          TARGET_BUCKET: gl-pdf-search-inter-store-bucket
      Layers:
        - 'arn:aws:lambda:us-east-1:032822161467:layer:pdf-search-aws-auth:1'
        - 'arn:aws:lambda:us-east-1:032822161467:layer:pdf-search-appbase:1'
      VpcConfig:
        SecurityGroupIds:
          - 'sg-0ac3ef597cdb9b3a6'
          - 'sg-e7437fb3'
        SubnetIds:
          - 'subnet-a8eef986'
          - 'subnet-740dd539'
          - 'subnet-13f4e14f'
  glPdfSearchDocumentStoreBucket:
    Type: 'AWS::S3::Bucket'
    Properties:
      BucketName: gl-pdf-search-document-store-bucket
      # The following NotificationConfiguration is used to trigger the
      # lambda function. But it cannot be used when the stack is created
      # for the first time. After the stack is created, the following
      # NotificationConfiguration can be added and the stack can be
      # updated. This will trigger the Lambda function when a new PDF
      # document is uploaded to the bucket
      # NotificationConfiguration:
      #   LambdaConfigurations:
      #     - Event: 's3:ObjectCreated:*'
      #       Function: !GetAtt glPdfSearchPdfToTxtLambda.Arn
  glPdfSearchDocumentStoreEventPdfToText:
    Type: 'AWS::Lambda::Permission'
    Properties:
      FunctionName: !Ref glPdfSearchPdfToTxtLambda
      Action: 'lambda:invokeFunction'
      Principal: s3.amazonaws.com
      SourceArn: !GetAtt glPdfSearchDocumentStoreBucket.Arn
```

buildspec.yml: This file is used by CodeBuild in the CodePipeline to package up the CloudFormation script that is deployed in the CodeDeploy step in the pipeline.

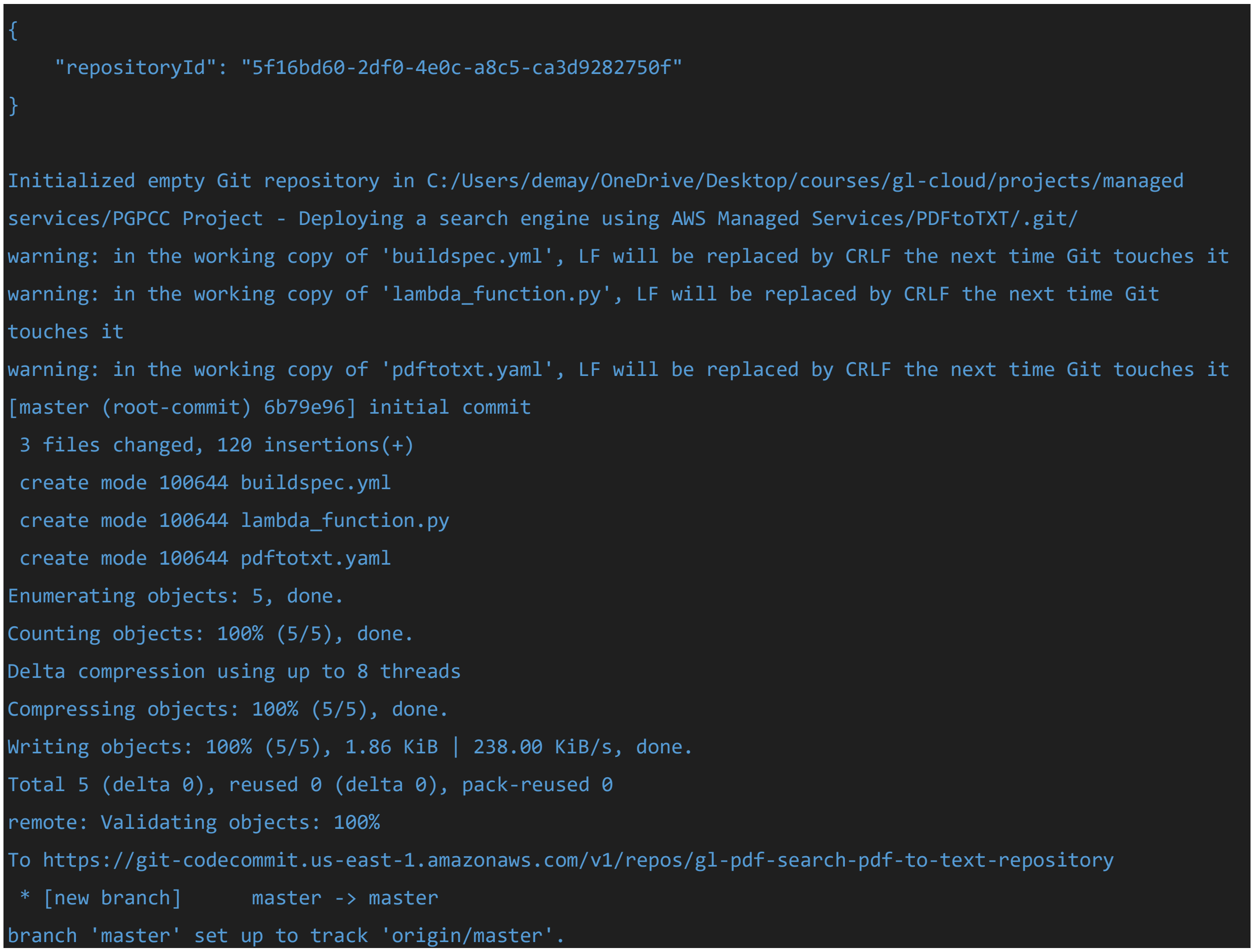
```
version: 0.2
phases:
  install:
    runtime-versions:
      python: 3.10
  build:
    commands:
      - aws cloudformation package --template-file pdftotxt.yaml --s3-bucket gl-pdf-search-artifacts-bucket --output-template-file lambdaoutput.yaml

artifacts:
  type: zip
  files:
    - lambdaoutput.yaml
```

recreate_repository.sh: I used this script to build my code repository.

```
#!/bin/bash

repository_name="gl-pdf-search-pdf-to-text-repository"
repository_description="GL Managed Services Project - PDFtoText"

# Clean up by removing the local repository
rm -rf .git

# Check if the repository already exists and delete it
aws codecommit get-repository --repository-name "$repository_name" > /dev/null 2>&1
if [ $? -eq 0 ]; then
    aws codecommit delete-repository --repository-name "$repository_name"
fi

# Create a new repository
clone_url_http=$(aws codecommit create-repository --repository-name "$repository_name" --repository-description "$repository_description" --output text --query 'repositoryMetadata.cloneUrlHttp')

# Initialize a local Git repository
git init
git add buildspec.yml lambda_function.py pdftotxt.yaml
git commit -m "initial commit"

# Add the remote repository and push the code
git remote add origin "$clone_url_http"
git push -u origin master
```

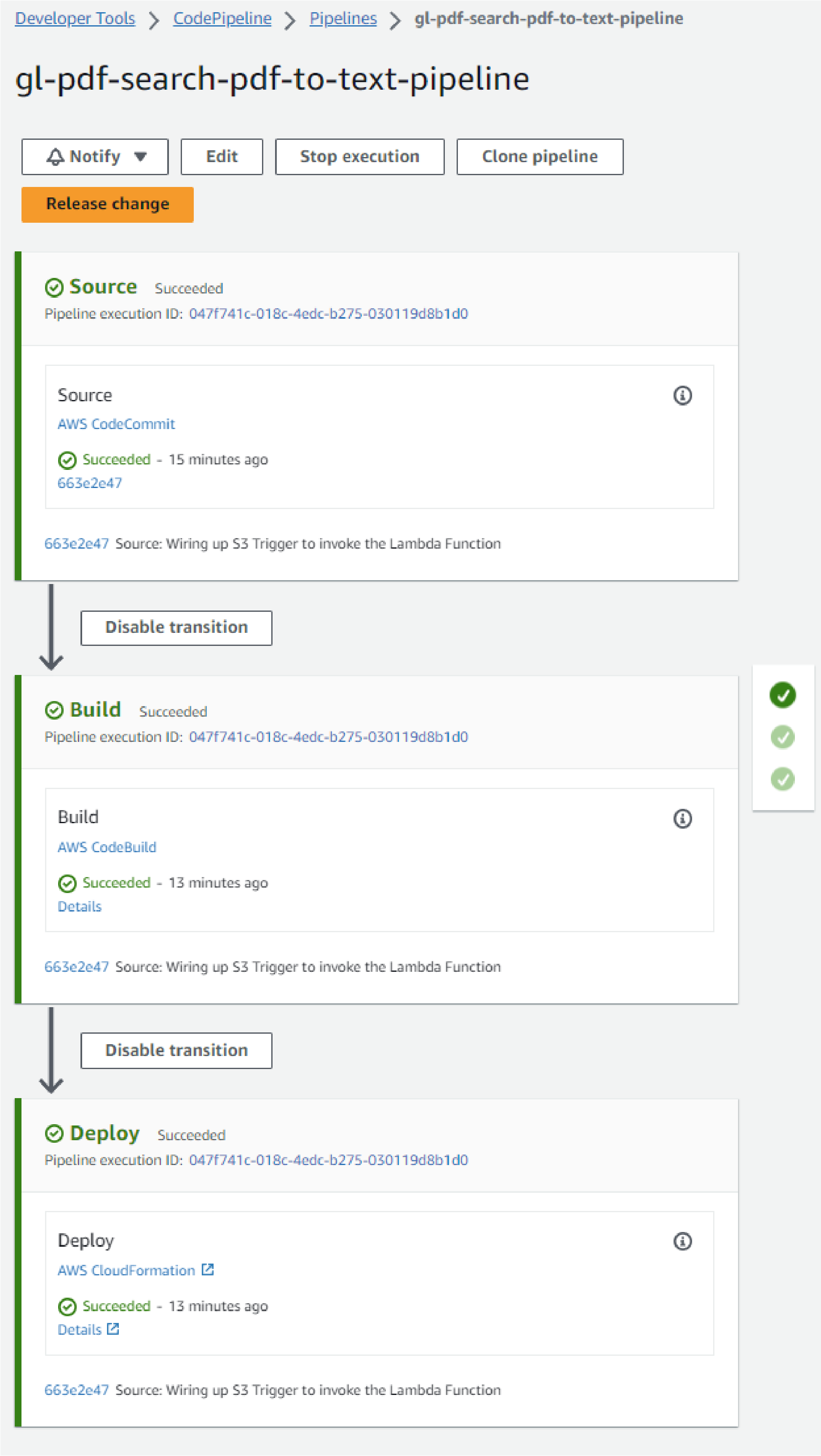
Output:

```
$ ./recreate_repository.sh
```



```
{
    "repositoryId": "5f16bd60-2df0-4e0c-a8c5-ca3d9282750f"
}

Initialized empty Git repository in C:/Users/demay/OneDrive/Desktop/courses/gl-cloud/projects/managed
services/PGPCC Project - Deploying a search engine using AWS Managed Services/PDFtoTXT/.git/
warning: in the working copy of 'buildspec.yml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'lambda_function.py', LF will be replaced by CRLF the next time Git
touches it
warning: in the working copy of 'pdftotxt.yaml', LF will be replaced by CRLF the next time Git touches it
[master (root-commit) 6b79e96] initial commit
 3 files changed, 120 insertions(+)
 create mode 100644 buildspec.yml
 create mode 100644 lambda_function.py
 create mode 100644 pdftotxt.yaml
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.86 KiB | 238.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.us-east-1.amazonaws.com/v1/repos/gl-pdf-search-pdf-to-text-repository
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

Next step. Setup the code pipeline for PDFtoText



gl-pdf-search-pdf-to-text-pipeline

✓ Source Succeeded
Pipeline execution ID: 047f741c-018c-4edc-b275-030119d8b1d0

Source
AWS CodeCommit
✓ Succeeded - 15 minutes ago
663e2e47

663e2e47 Source: Wiring up S3 Trigger to invoke the Lambda Function

✓ Build Succeeded
Pipeline execution ID: 047f741c-018c-4edc-b275-030119d8b1d0

Build
AWS CodeBuild
✓ Succeeded - 13 minutes ago
Details

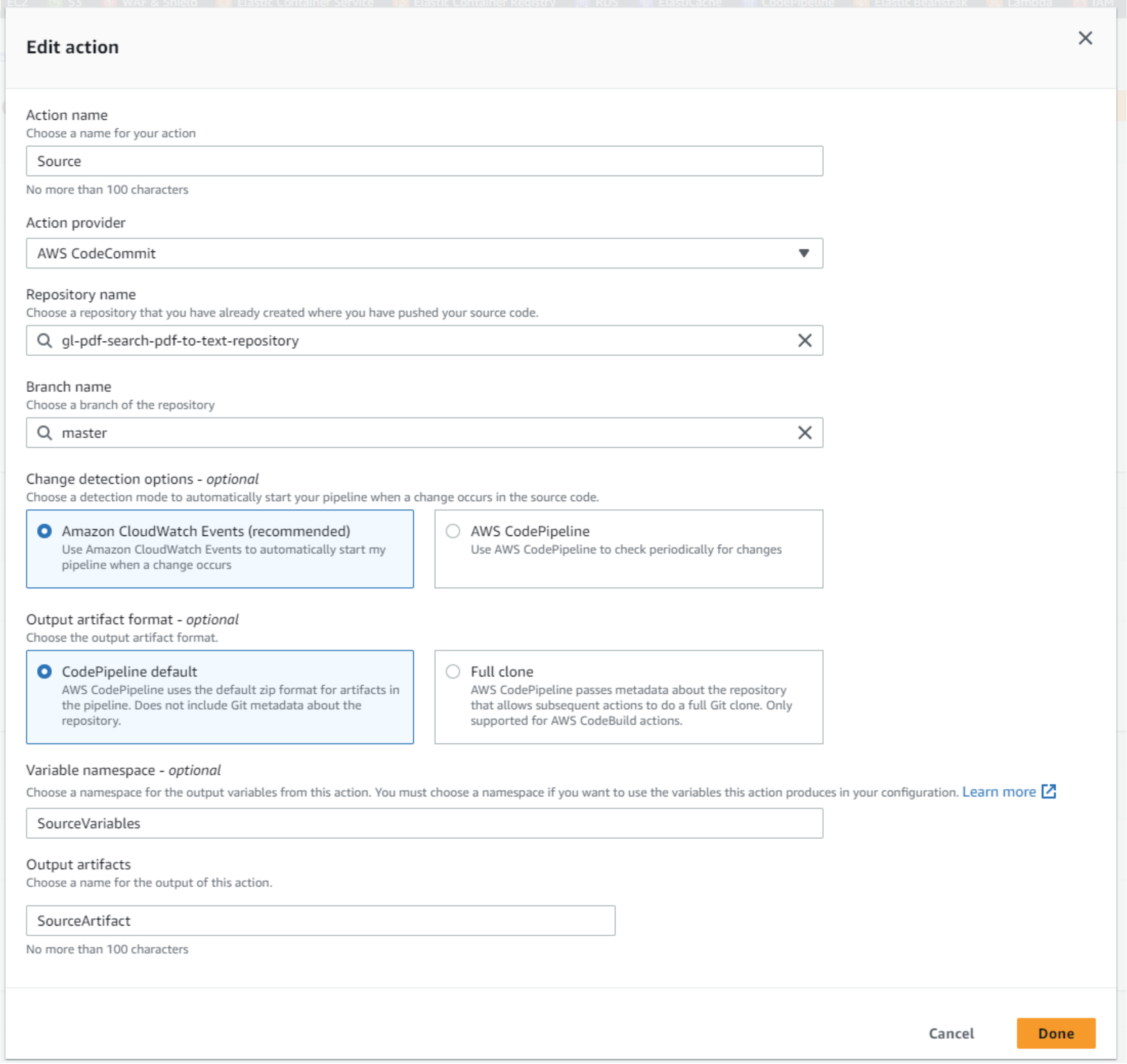
663e2e47 Source: Wiring up S3 Trigger to invoke the Lambda Function

✓ Deploy Succeeded
Pipeline execution ID: 047f741c-018c-4edc-b275-030119d8b1d0

Deploy
AWS CloudFormation
✓ Succeeded - 13 minutes ago
Details

663e2e47 Source: Wiring up S3 Trigger to invoke the Lambda Function

Source step: This is triggered whenever a new commit is made to `gl-pdf-search-pdf-to-text-repository`



Edit action

Action name
Choose a name for your action
Source
No more than 100 characters

Action provider
AWS CodeCommit

Repository name
Choose a repository that you have already created where you have pushed your source code.
gl-pdf-search-pdf-to-text-repository

Branch name
Choose a branch of the repository
master

Change detection options - *optional*
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

● Amazon CloudWatch Events (recommended)
Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs

○ AWS CodePipeline
Use AWS CodePipeline to check periodically for changes

Output artifact format - *optional*
Choose the output artifact format.

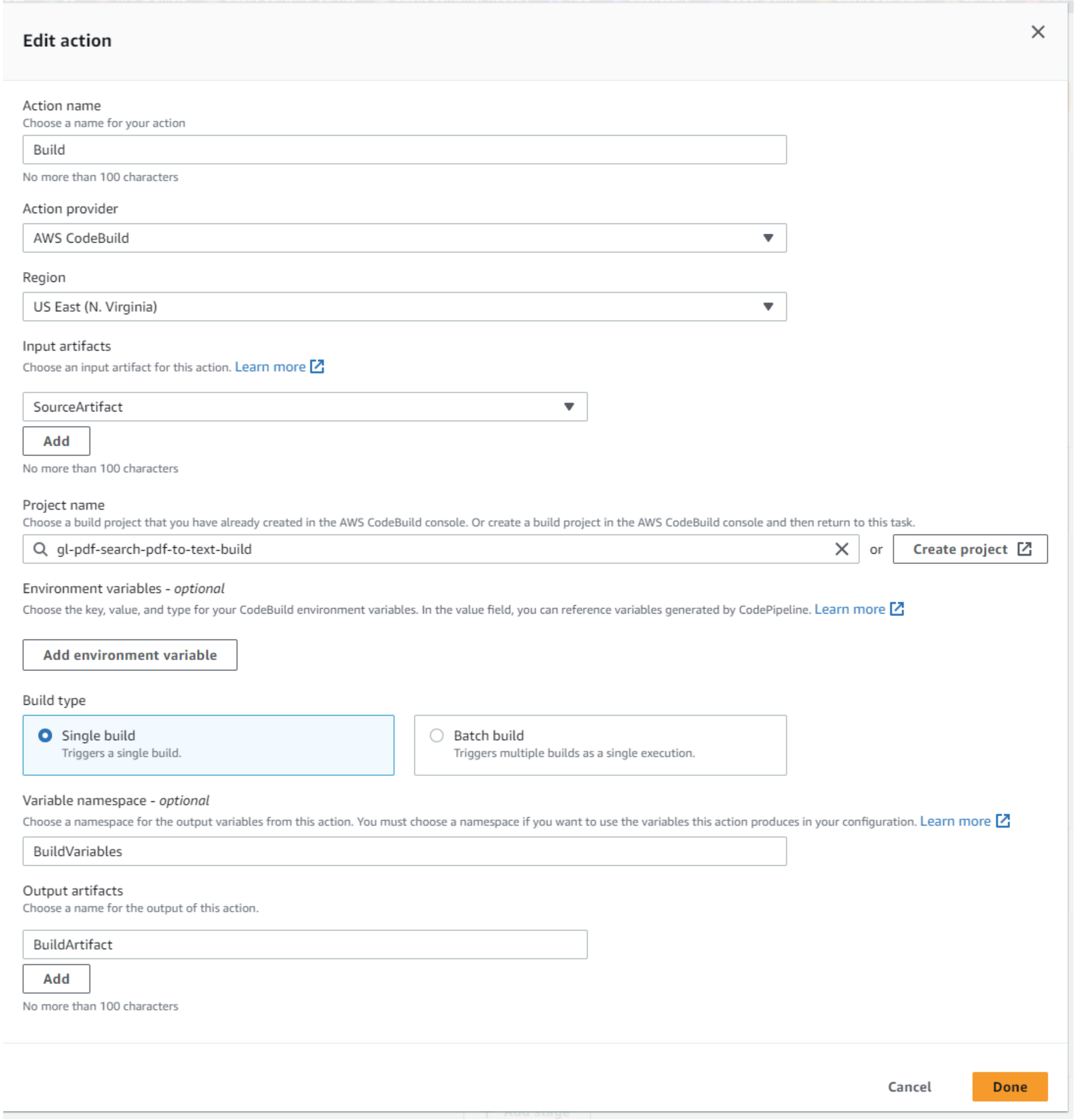
● CodePipeline default
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.

○ Full clone
AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only supported for AWS CodeBuild actions.

Variable namespace - *optional*
Choose a namespace for the output variables from this action. You must choose a namespace if you want to use the variables this action produces in your configuration. Learn more
SourceVariables

Output artifacts
Choose a name for the output of this action.
SourceArtifact
No more than 100 characters

Build step: This calls `aws cloudformation package` to package everything up to deploy to CloudFormation.



Edit action

Action name
Choose a name for your action
Build
No more than 100 characters

Action provider
AWS CodeBuild

Region
US East (N. Virginia)

Input artifacts
Choose an input artifact for this action. Learn more
SourceArtifact
No more than 100 characters

Project name
Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.
gl-pdf-search-pdf-to-text-build

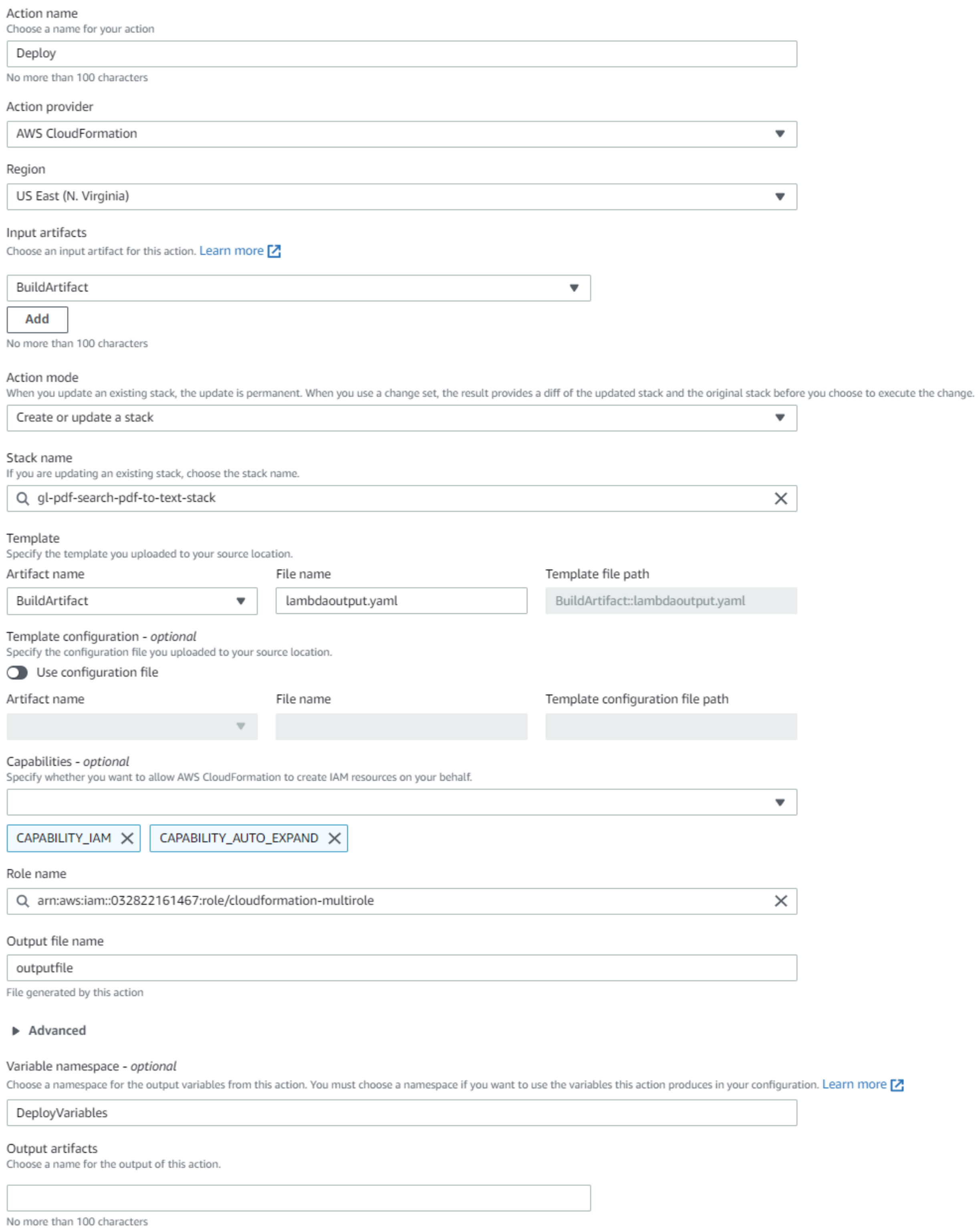
Environment variables - *optional*
Choose the key, value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. Learn more

Build type
● Single build
Triggers a single build.
○ Batch build
Triggers multiple builds as a single execution.

Variable namespace - *optional*
Choose a namespace for the output variables from this action. You must choose a namespace if you want to use the variables this action produces in your configuration. Learn more
BuildVariables

Output artifacts
Choose a name for the output of this action.
BuildArtifact
No more than 100 characters

Deploy Stage: After everything is built, the package is pushed out to cloudformation.



Action name

Choose a name for your action

Deploy

No more than 100 characters

Action provider

AWS CloudFormation

Region

US East (N. Virginia)

Input artifacts

Choose an input artifact for this action. Learn more

BuildArtifact

Add

No more than 100 characters

Action mode

When you update an existing stack, the update is permanent. When you use a change set, the result provides a diff of the updated stack and the original stack before you choose to execute the change.

Create or update a stack

Stack name

If you are updating an existing stack, choose the stack name.

gl-pdf-search-pdf-to-text-stack

Template

Specify the template you uploaded to your source location.

Artifact name	File name	Template file path
BuildArtifact	lambdaoutput.yaml	BuildArtifact::lambdaoutput.yaml

Template configuration - *optional*

Specify the configuration file you uploaded to your source location.

○ Use configuration file

Artifact name	File name	Template configuration file path

Capabilities - *optional*

Specify whether you want to allow AWS CloudFormation to create IAM resources on your behalf.

CAPABILITY_IAM ✕ CAPABILITY_AUTO_EXPAND ✕

Role name

arn:aws:iam::032822161467:role/cloudformation-multirole

Output file name

outputfile

File generated by this action

▶ Advanced

Variable namespace - *optional*

Choose a namespace for the output variables from this action. You must choose a namespace if you want to use the variables this action produces in your configuration. Learn more

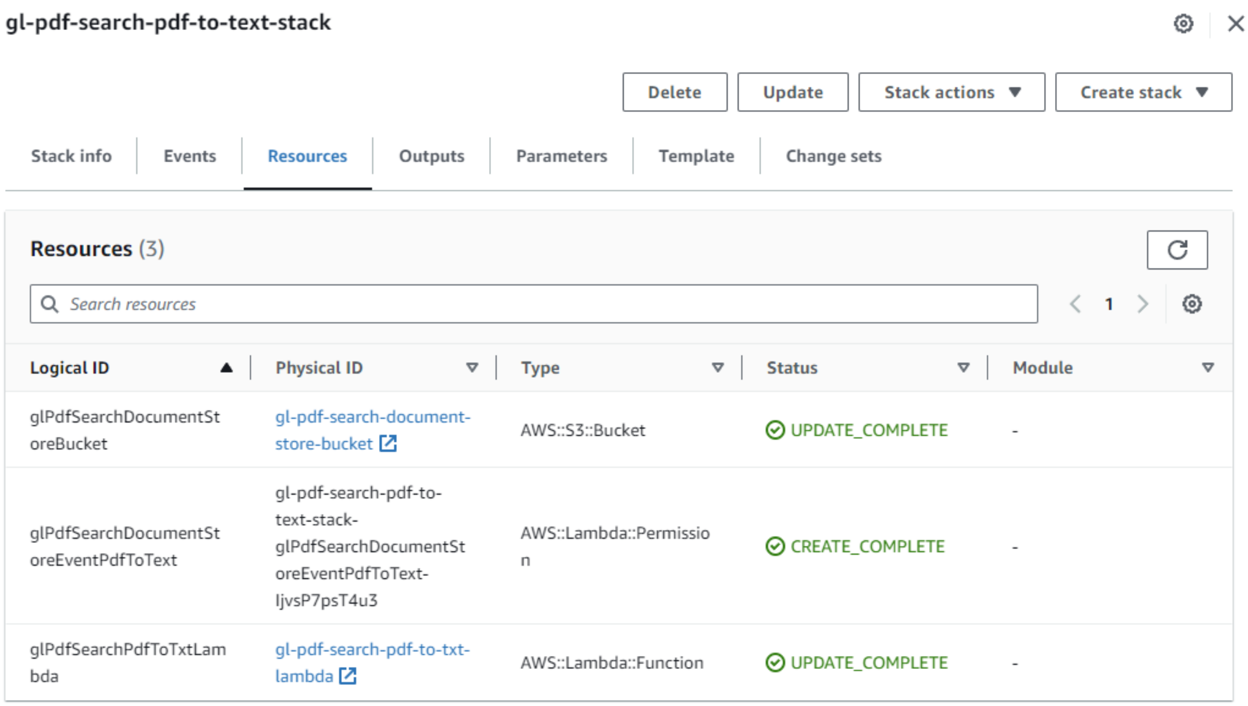
DeployVariables

Output artifacts

Choose a name for the output of this action.

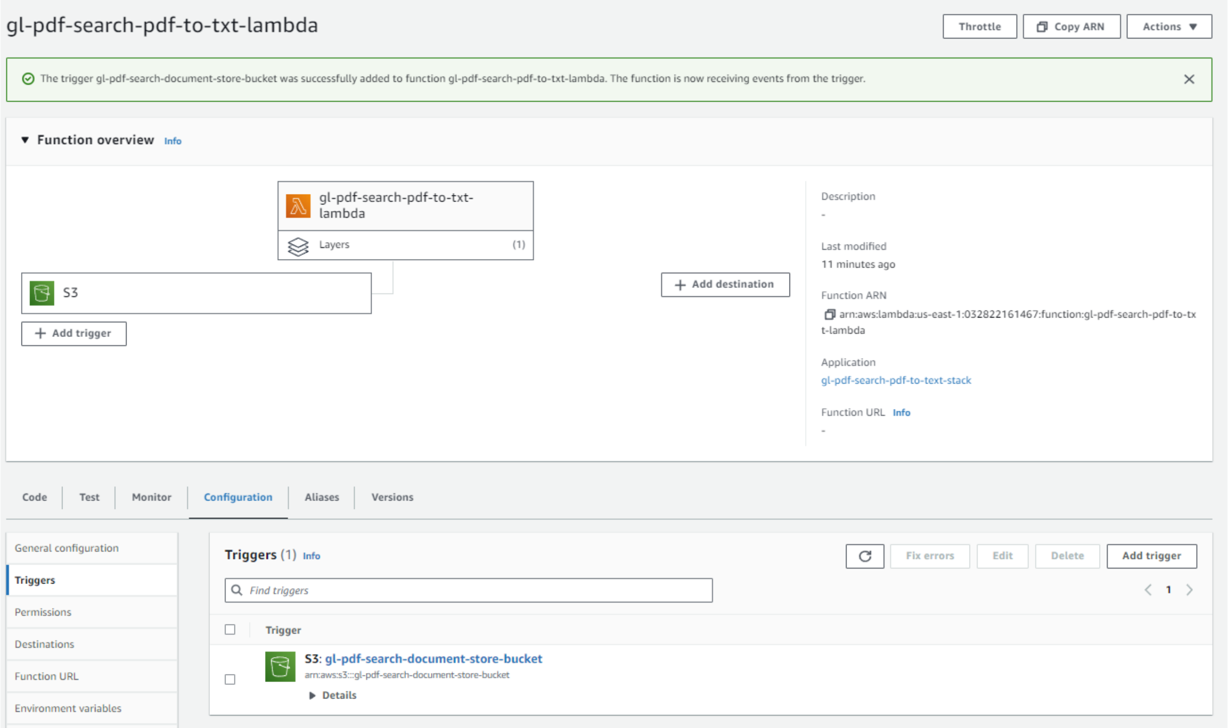
No more than 100 characters

These are the resources that are created the second time code is pushed through the pipeline (event has to be added after stack initial creation)



Logical ID	Physical ID	Type	Status	Module
glPdfSearchDocumentStoreBucket	gl-pdf-search-document-store-bucket	AWS::S3::Bucket	UPDATE_COMPLETE	-
glPdfSearchDocumentStoreEventPdfToText	gl-pdf-search-pdf-to-text-stack-glPdfSearchDocumentStoreEventPdfToText-ljvsP7psT4u3	AWS::Lambda::Permission	CREATE_COMPLETE	-
glPdfSearchPdfToTxtLambda	gl-pdf-search-pdf-to-txt-lambda	AWS::Lambda::Function	UPDATE_COMPLETE	-

The lambda function is created by the CloudFormation stack.



The trigger gl-pdf-search-document-store-bucket was successfully added to function gl-pdf-search-pdf-to-txt-lambda. The function is now receiving events from the trigger.

Description
-

Last modified
11 minutes ago

Function ARN
arn:aws:lambda:us-east-1:032822161467:function:gl-pdf-search-pdf-to-txt-lambda

Application
gl-pdf-search-pdf-to-text-stack

Function URL
-

S3: gl-pdf-search-document-store-bucket
arn:aws:s3:::gl-pdf-search-document-store-bucket

UploadToSearch: The following lambda function is used to take the extracted PDF text from PDFToText lambda function and upload it to AWS OpenSearch.

Basically everything done for PDFToText was repeated for UploadToSearch with the following exceptions.

- The pypdf lambda layer is not needed
- A VPC Endpoint needed to be added so the lambda function can reach AWS Opensearch domain

buildspec.yml

```
version: 0.2
phases:
  install:
    runtime-versions:
      python: 3.10
  build:
    commands:
      - aws cloudformation package --template-file upload-to-search.yaml --s3-bucket gl-pdf-search-artifacts-bucket --output-template-file upload.yaml

artifacts:
  type: zip
  files:
    - upload.yaml
```

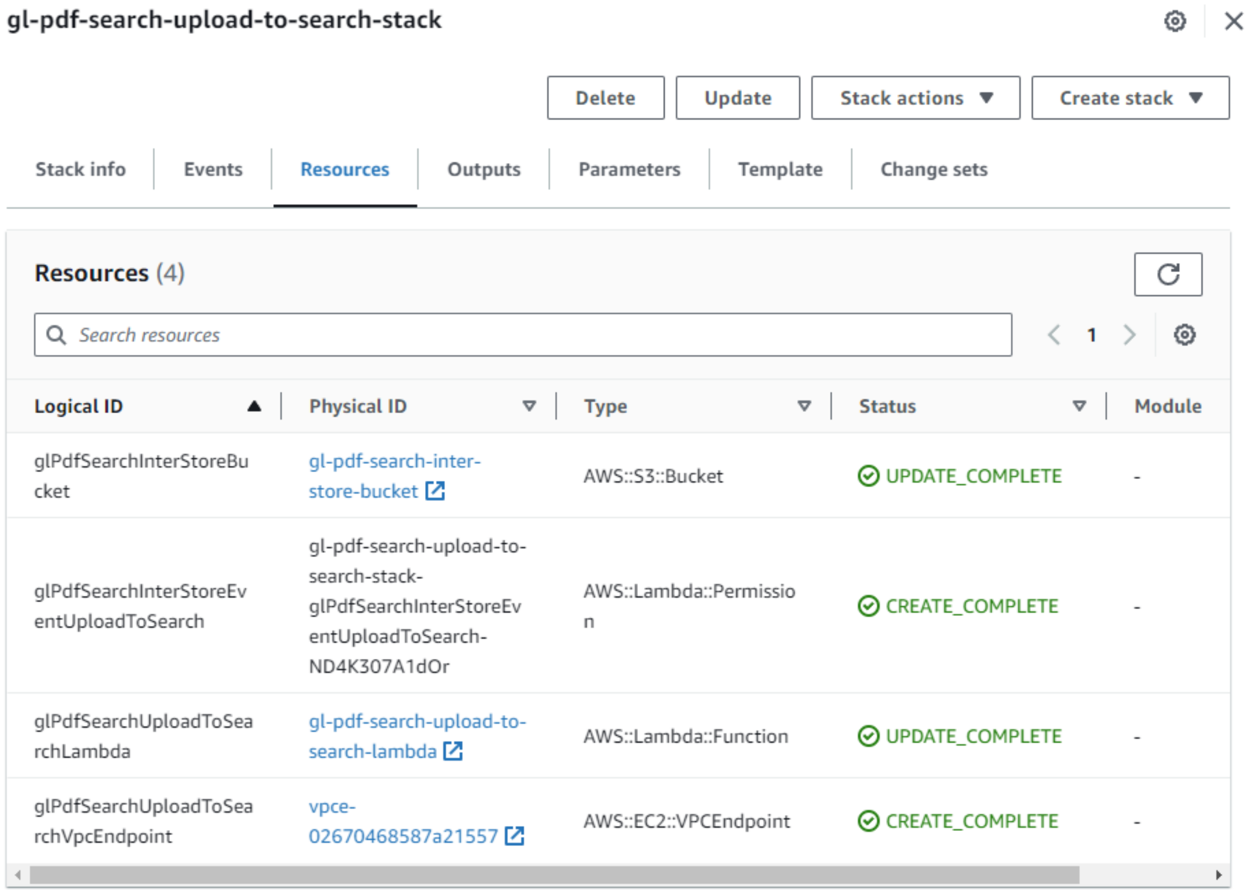
upload-to-search.yaml:

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: 'AWS::Serverless-2016-10-31'
Description: An AWS Serverless Specification template describing your function.
Resources:
  glPdfSearchUploadToSearchLambda:
    Type: 'AWS::Serverless::Function'
    Properties:
      FunctionName: gl-pdf-search-upload-to-search-lambda
      Handler: lambda_function.handler
      Runtime: python3.9
      CodeUri: .
      Description: ''
      MemorySize: 512
      Timeout: 900
      Role: 'arn:aws:iam::032822161467:role/lambda-multirole'
      Layers:
        - 'arn:aws:lambda:us-east-1:032822161467:layer:pdf-search-aws-auth:1'
      VpcConfig:
        SecurityGroupIds:
```

```yaml
        - 'sg-0ac3ef597cdb9b3a6'
        - 'sg-e7437fb3'
      SubnetIds:
        - 'subnet-a8eef986'
        - 'subnet-740dd539'
        - 'subnet-13f4e14f'
glPdfSearchInterStoreBucket:
  Type: 'AWS::S3::Bucket'
  Properties:
    BucketName: gl-pdf-search-inter-store-bucket
    # The following NotificationConfiguration is used to trigger the
    # lambda function. But it cannot be used when the stack is created
    # for the first time. After the stack is created, the following
    # NotificationConfiguration can be added and the stack can be
    # updated. This will trigger the Lambda function when a new PDF
    # document is uploaded to the bucket
    NotificationConfiguration:
      LambdaConfigurations:
        - Event: 's3:ObjectCreated:*'
          Function: !GetAtt glPdfSearchUploadToSearchLambda.Arn
glPdfSearchUploadToSearchVpcEndpoint:
  Type: 'AWS::EC2::VPCEndpoint'
  Properties:
    ServiceName: com.amazonaws.us-east-1.s3
    VpcEndpointType: Gateway
    VpcId: "vpc-2a735750"
    RouteTableIds: ["rtb-ecf65592"]
    PolicyDocument:
      Version: "2008-10-17"
      Statement:
        - Effect: "Allow"
          Principal: "*"
          Action: "*"
          Resource: "*"
glPdfSearchInterStoreEventUploadToSearch:
  Type: 'AWS::Lambda::Permission'
  Properties:
    FunctionName: !Ref glPdfSearchUploadToSearchLambda
    Action: 'lambda:invokeFunction'
    Principal: s3.amazonaws.com
    SourceArn: arn:aws:s3:::gl-pdf-search-inter-store-bucket
```

The repository and code pipeline was created in the same manner as PDFToText. Pretty much identical except for the resource names. The screen shots have been omitted for brevity.

These are the resources created by the CloudFormation stack

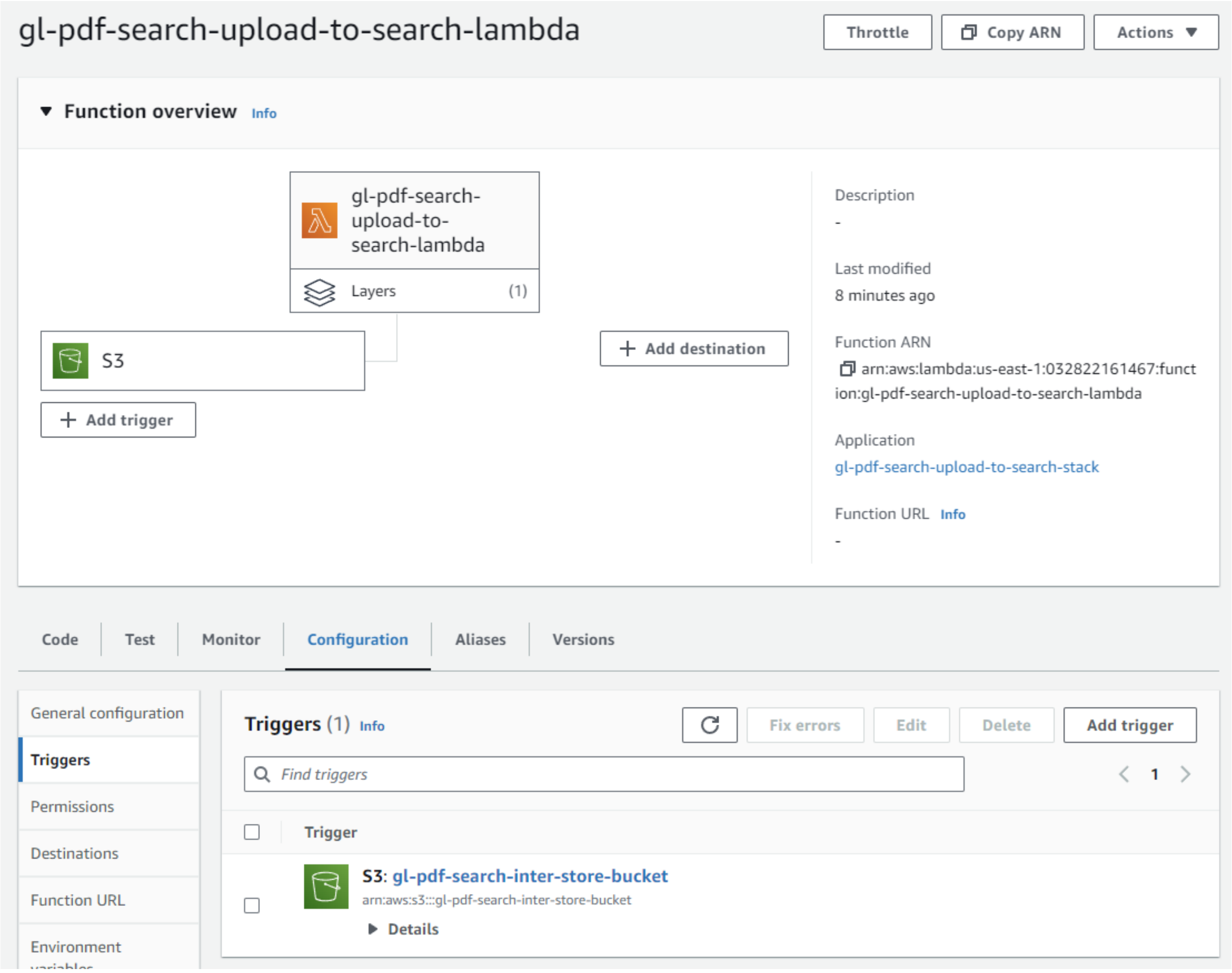


gl-pdf-search-upload-to-search-stack

Resources (4)

Logical ID	Physical ID	Type	Status	Module
glPdfSearchInterStoreBucket	gl-pdf-search-inter-store-bucket	AWS::S3::Bucket	✓ UPDATE_COMPLETE	-
glPdfSearchInterStoreEventUploadToSearch	gl-pdf-search-upload-to-search-stack-glPdfSearchInterStoreEventUploadToSearch-ND4K307A1dOr	AWS::Lambda::Permission	✓ CREATE_COMPLETE	-
glPdfSearchUploadToSearchLambda	gl-pdf-search-upload-to-search-lambda	AWS::Lambda::Function	✓ UPDATE_COMPLETE	-
glPdfSearchUploadToSearchVpcEndpoint	vpce-02670468587a21557	AWS::EC2::VPCEndpoint	✓ CREATE_COMPLETE	-

The lambda function created by the CloudFormation stack:



gl-pdf-search-upload-to-search-lambda

Description
-

Last modified
8 minutes ago

Function ARN
arn:aws:lambda:us-east-1:032822161467:function:gl-pdf-search-upload-to-search-lambda

Application
gl-pdf-search-upload-to-search-stack

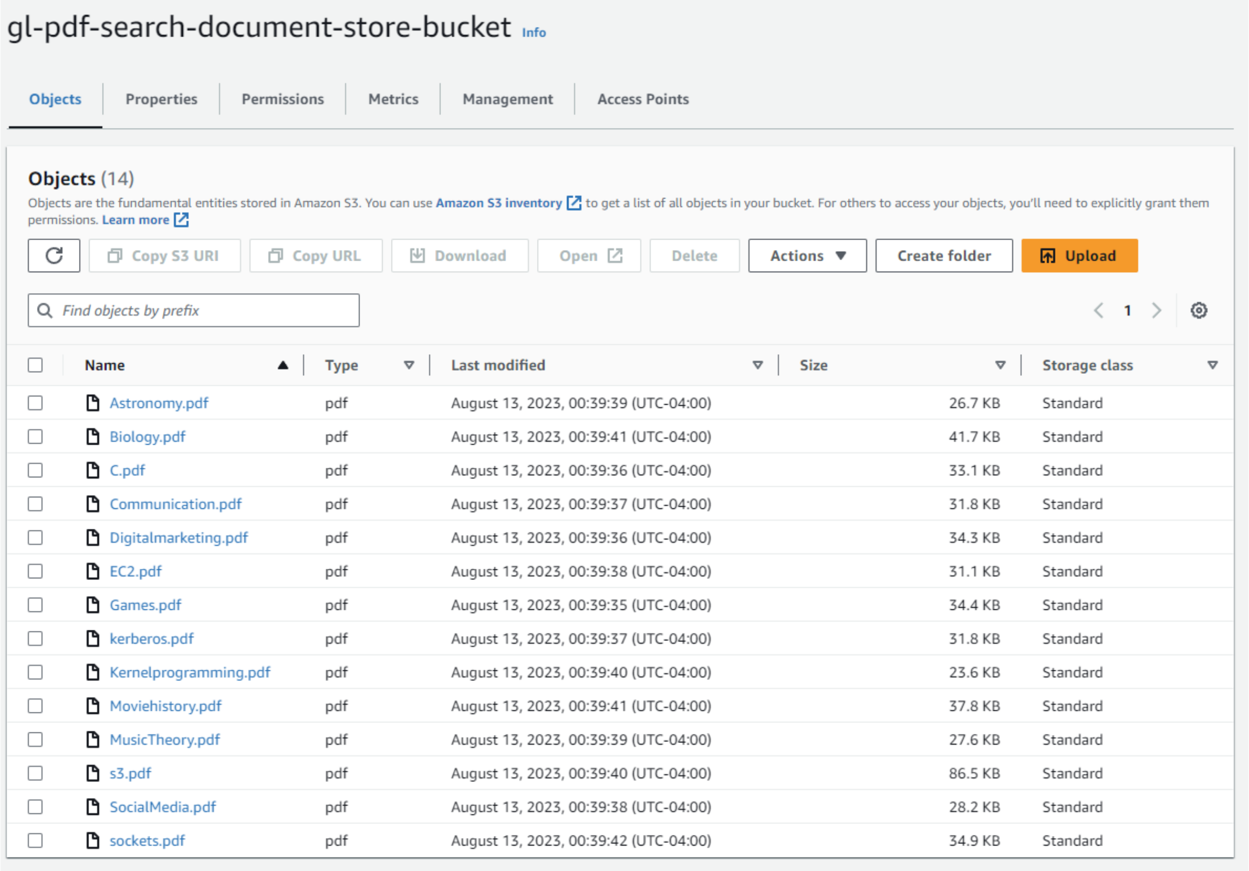
Function URL
-

Triggers (1)

S3: gl-pdf-search-inter-store-bucket
arn:aws:s3:::gl-pdf-search-inter-store-bucket

Executing the first two lambda functions:

Uploading PDFs for the document store bucket.



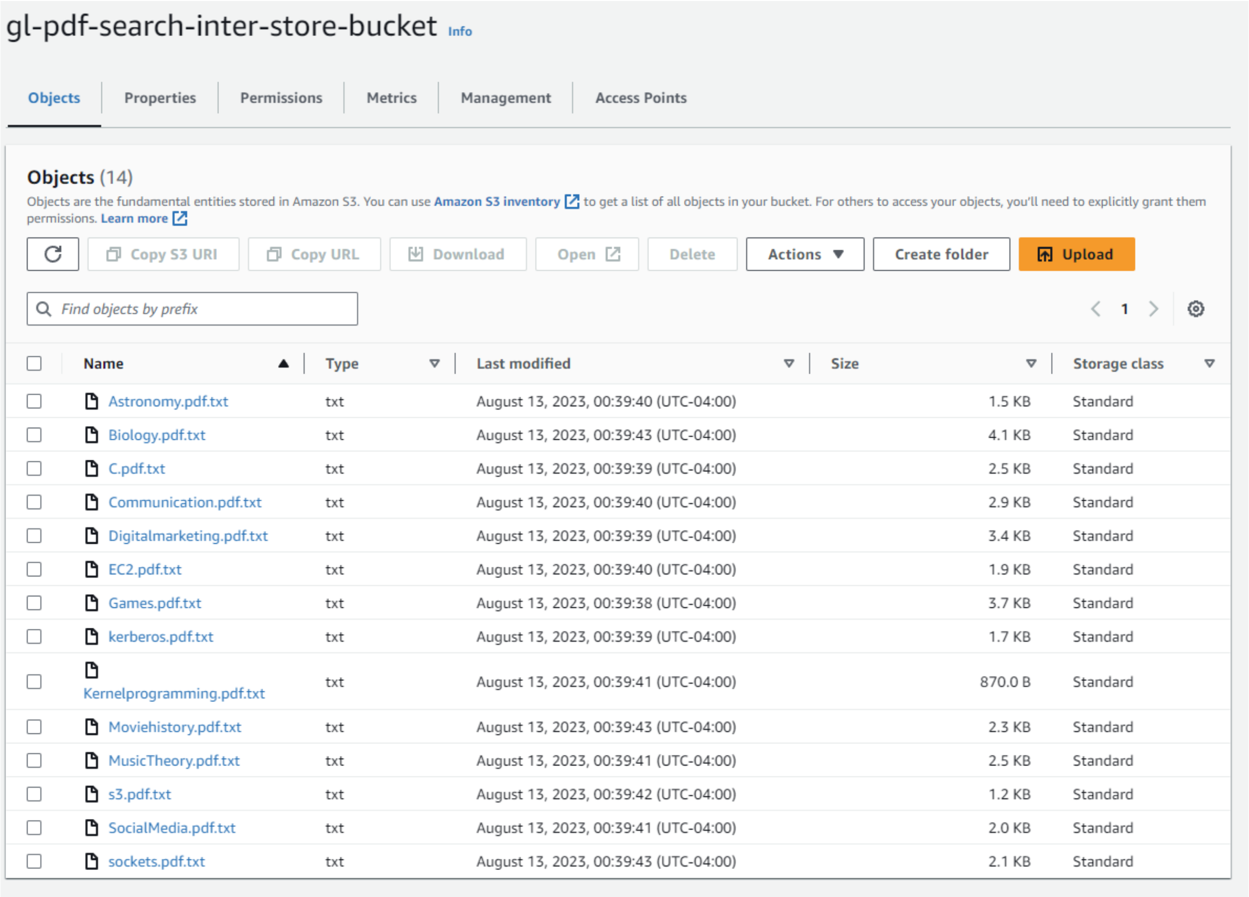
gl-pdf-search-document-store-bucket

Objects (14)

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

□	Name	Type	Last modified	Size	Storage class
□	Astronomy.pdf	pdf	August 13, 2023, 00:39:39 (UTC-04:00)	26.7 KB	Standard
□	Biology.pdf	pdf	August 13, 2023, 00:39:41 (UTC-04:00)	41.7 KB	Standard
□	C.pdf	pdf	August 13, 2023, 00:39:36 (UTC-04:00)	33.1 KB	Standard
□	Communication.pdf	pdf	August 13, 2023, 00:39:37 (UTC-04:00)	31.8 KB	Standard
□	Digitalmarketing.pdf	pdf	August 13, 2023, 00:39:36 (UTC-04:00)	34.3 KB	Standard
□	EC2.pdf	pdf	August 13, 2023, 00:39:38 (UTC-04:00)	31.1 KB	Standard
□	Games.pdf	pdf	August 13, 2023, 00:39:35 (UTC-04:00)	34.4 KB	Standard
□	kerberos.pdf	pdf	August 13, 2023, 00:39:37 (UTC-04:00)	31.8 KB	Standard
□	Kernelprogramming.pdf	pdf	August 13, 2023, 00:39:40 (UTC-04:00)	23.6 KB	Standard
□	Moviehistory.pdf	pdf	August 13, 2023, 00:39:41 (UTC-04:00)	37.8 KB	Standard
□	MusicTheory.pdf	pdf	August 13, 2023, 00:39:39 (UTC-04:00)	27.6 KB	Standard
□	s3.pdf	pdf	August 13, 2023, 00:39:40 (UTC-04:00)	86.5 KB	Standard
□	SocialMedia.pdf	pdf	August 13, 2023, 00:39:38 (UTC-04:00)	28.2 KB	Standard
□	sockets.pdf	pdf	August 13, 2023, 00:39:42 (UTC-04:00)	34.9 KB	Standard

Text extracted from the PDF files by the PDFtoText lambda function and uploaded to intermediary storage bucket

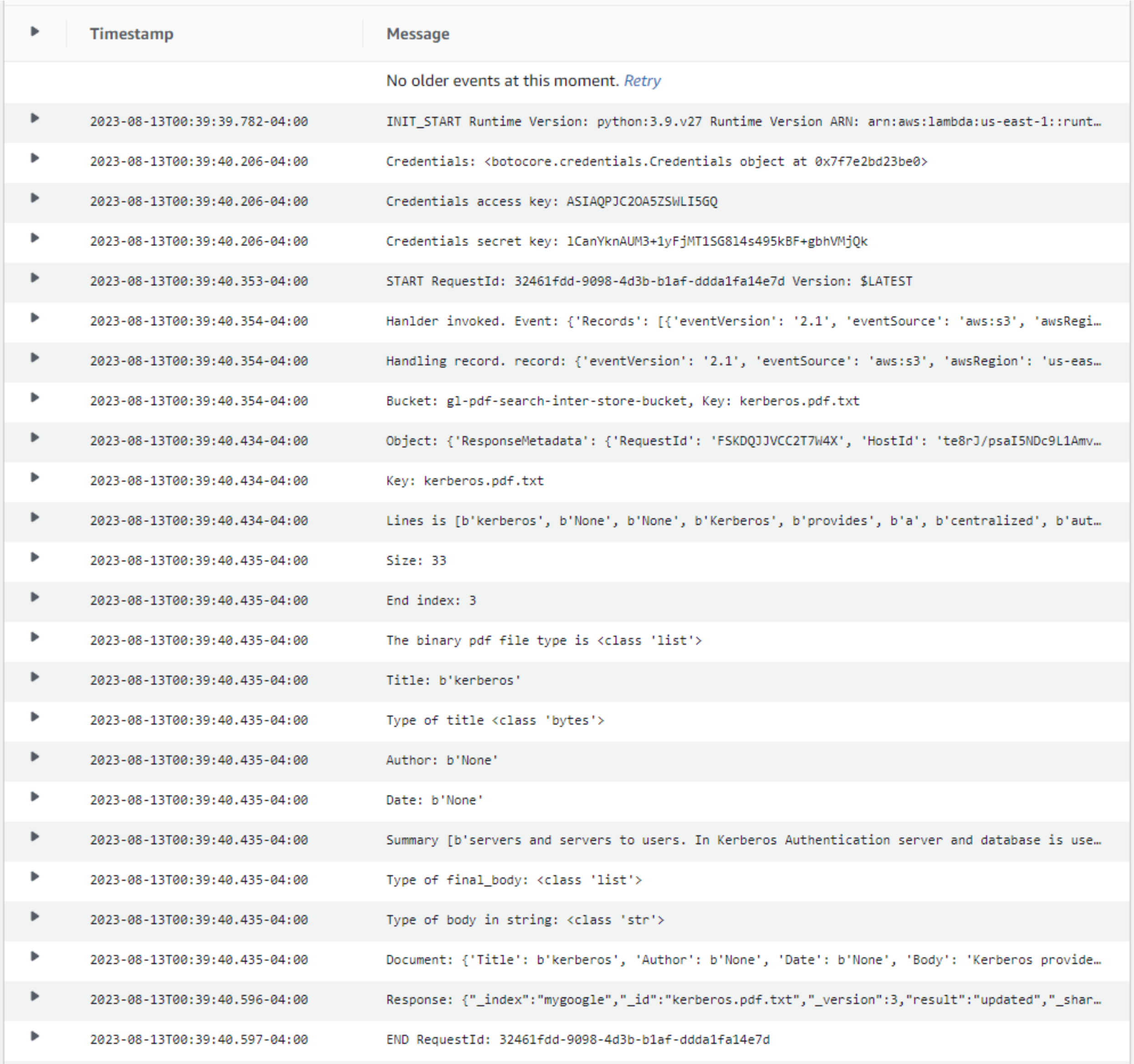


Objects (14)

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

□	Name ▲	Type	Last modified	Size	Storage class
□	Astronomy.pdf.txt	txt	August 13, 2023, 00:39:40 (UTC-04:00)	1.5 KB	Standard
□	Biology.pdf.txt	txt	August 13, 2023, 00:39:43 (UTC-04:00)	4.1 KB	Standard
□	C.pdf.txt	txt	August 13, 2023, 00:39:39 (UTC-04:00)	2.5 KB	Standard
□	Communication.pdf.txt	txt	August 13, 2023, 00:39:40 (UTC-04:00)	2.9 KB	Standard
□	Digitalmarketing.pdf.txt	txt	August 13, 2023, 00:39:39 (UTC-04:00)	3.4 KB	Standard
□	EC2.pdf.txt	txt	August 13, 2023, 00:39:40 (UTC-04:00)	1.9 KB	Standard
□	Games.pdf.txt	txt	August 13, 2023, 00:39:38 (UTC-04:00)	3.7 KB	Standard
□	kerberos.pdf.txt	txt	August 13, 2023, 00:39:39 (UTC-04:00)	1.7 KB	Standard
□	Kernelprogramming.pdf.txt	txt	August 13, 2023, 00:39:41 (UTC-04:00)	870.0 B	Standard
□	Moviehistory.pdf.txt	txt	August 13, 2023, 00:39:43 (UTC-04:00)	2.3 KB	Standard
□	MusicTheory.pdf.txt	txt	August 13, 2023, 00:39:41 (UTC-04:00)	2.5 KB	Standard
□	s3.pdf.txt	txt	August 13, 2023, 00:39:42 (UTC-04:00)	1.2 KB	Standard
□	SocialMedia.pdf.txt	txt	August 13, 2023, 00:39:41 (UTC-04:00)	2.0 KB	Standard
□	sockets.pdf.txt	txt	August 13, 2023, 00:39:43 (UTC-04:00)	2.1 KB	Standard

CloudWatch logs showing the output from the UploadToSearch lambda function.



```
    Timestamp                        Message
                                     No older events at this moment. Retry
    2023-08-13T00:39:39.782-04:00    INIT_START Runtime Version: python:3.9.v27 Runtime Version ARN: arn:aws:lambda:us-east-1::runt…
    2023-08-13T00:39:40.206-04:00    Credentials: <botocore.credentials.Credentials object at 0x7f7e2bd23be0>
    2023-08-13T00:39:40.206-04:00    Credentials access key: ASIAQPJC2OA5ZSWLI5GQ
    2023-08-13T00:39:40.206-04:00    Credentials secret key: 1CanYknAUM3+1yFjMT1SG814s495kBF+gbhVMjQk
    2023-08-13T00:39:40.353-04:00    START RequestId: 32461fdd-9098-4d3b-b1af-ddda1fa14e7d Version: $LATEST
    2023-08-13T00:39:40.354-04:00    Hanlder invoked. Event: {'Records': [{'eventVersion': '2.1', 'eventSource': 'aws:s3', 'awsRegi…
    2023-08-13T00:39:40.354-04:00    Handling record. record: {'eventVersion': '2.1', 'eventSource': 'aws:s3', 'awsRegion': 'us-eas…
    2023-08-13T00:39:40.354-04:00    Bucket: gl-pdf-search-inter-store-bucket, Key: kerberos.pdf.txt
    2023-08-13T00:39:40.434-04:00    Object: {'ResponseMetadata': {'RequestId': 'FSKDQJJVCC2T7W4X', 'HostId': 'te8rJ/psaI5NDc9L1Amv…
    2023-08-13T00:39:40.434-04:00    Key: kerberos.pdf.txt
    2023-08-13T00:39:40.434-04:00    Lines is [b'kerberos', b'None', b'None', b'Kerberos', b'provides', b'a', b'centralized', b'aut…
    2023-08-13T00:39:40.435-04:00    Size: 33
    2023-08-13T00:39:40.435-04:00    End index: 3
    2023-08-13T00:39:40.435-04:00    The binary pdf file type is <class 'list'>
    2023-08-13T00:39:40.435-04:00    Title: b'kerberos'
    2023-08-13T00:39:40.435-04:00    Type of title <class 'bytes'>
    2023-08-13T00:39:40.435-04:00    Author: b'None'
    2023-08-13T00:39:40.435-04:00    Date: b'None'
    2023-08-13T00:39:40.435-04:00    Summary [b'servers and servers to users. In Kerberos Authentication server and database is use…
    2023-08-13T00:39:40.435-04:00    Type of final_body: <class 'list'>
    2023-08-13T00:39:40.435-04:00    Type of body in string: <class 'str'>
    2023-08-13T00:39:40.435-04:00    Document: {'Title': b'kerberos', 'Author': b'None', 'Date': b'None', 'Body': 'Kerberos provide…
    2023-08-13T00:39:40.596-04:00    Response: {"_index":"mygoogle","_id":"kerberos.pdf.txt","_version":3,"result":"updated","_shar…
    2023-08-13T00:39:40.597-04:00    END RequestId: 32461fdd-9098-4d3b-b1af-ddda1fa14e7d
```

Now that we have the text extracted from the PDFs and Uploaded to the Opensearch domain, the next step is to create the user experience.

SearchGateway: This lambda function serves the “Search Page” to the user.

buildspec.yml

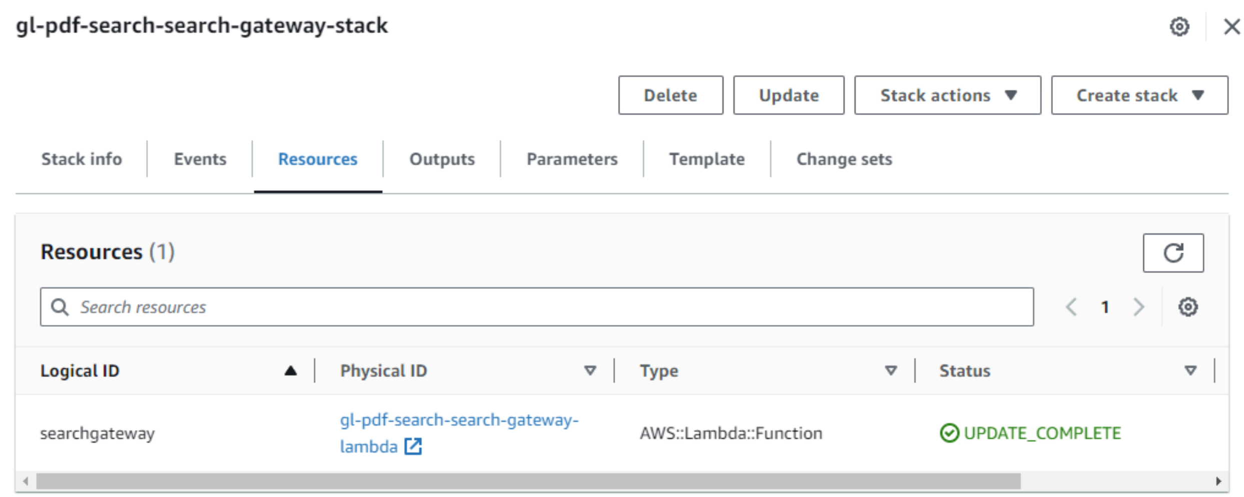
```
version: 0.2
phases:
  install:
    runtime-versions:
      python: 3.10
  build:
    commands:
      - aws cloudformation package --template-file search-gateway.yaml --s3-bucket gl-pdf-search-artifacts-bucket --output-template-file search.yaml

artifacts:
  type: zip
  files:
    - search.yaml
```

search-gateway.yaml. The CloudFormation template that creates the lambda function. No S3 buckets or event triggers are needed for this. An event trigger from an API HTTP Gateway is added later.

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: 'AWS::Serverless-2016-10-31'
Description: >-
  Process the incoming HTTP requests from the API Gateway to run search
  queries on the Opensearch domain
Resources:
  searchgateway:
    Type: 'AWS::Serverless::Function'
    Properties:
      FunctionName: gl-pdf-search-search-gateway-lambda
      Handler: lambda_function.lambda_handler
      Runtime: python3.9
      CodeUri: .
      Description: ''
      MemorySize: 128
      Timeout: 300
      Role: 'arn:aws:iam::032822161467:role/lambda-multirole'
      Layers:
        - 'arn:aws:lambda:us-east-1:032822161467:layer:pdf-search-aws-auth:1'
      VpcConfig:
        SecurityGroupIds:
          - 'sg-0ac3ef597cdb9b3a6'
          - 'sg-e7437fb3'
        SubnetIds:
          - 'subnet-a8eef986'
          - 'subnet-740dd539'
          - 'subnet-13f4e14f'
```

The repository and code pipeline was created in the same manner as the previous two lambda functions. Everything is identical except for the resource names. The screen shots have been omitted for brevity.

The resources created in the CloudFormation stack



gl-pdf-search-search-gateway-stack

Resources (1)

Logical ID ▲	Physical ID	Type	Status
searchgateway	gl-pdf-search-search-gateway-lambda	AWS::Lambda::Function	✓ UPDATE_COMPLETE

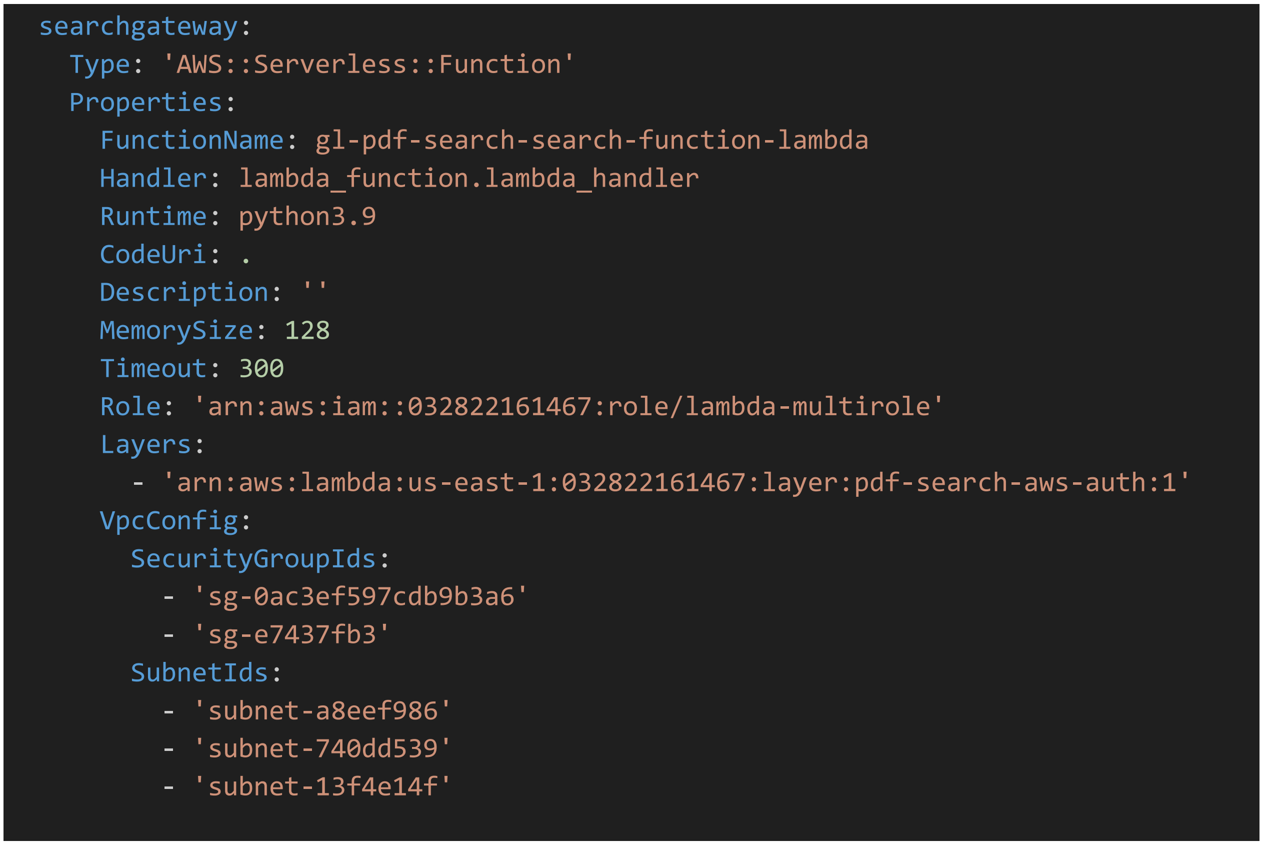
SearchFunction: This lambda function handles requests submitted from the “Search Page” implemented in the SearchGateway lambda function.

buildspec.yml

```
version: 0.2
phases:
  install:
    runtime-versions:
      python: 3.10
  build:
    commands:
      - aws cloudformation package --template-file search-function.yaml --s3-bucket gl-pdf-search-artifacts-bucket --output-template-file function.yaml
artifacts:
  type: zip
  files:
    - function.yaml
```

search-function.yaml: The CloudFormation template that creates the lambda function. No S3 buckets or event triggers are needed here for this. An event trigger from an API HTTP gateway is added later.

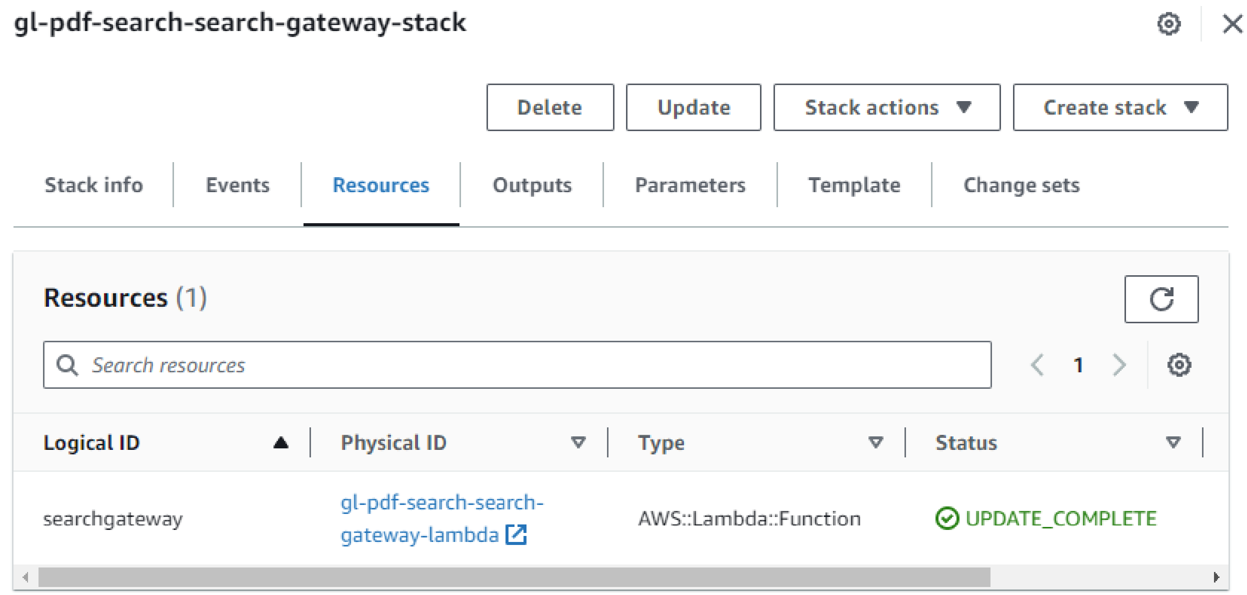
```
AWSTemplateFormatVersion: '2010-09-09'
Transform: 'AWS::Serverless-2016-10-31'
Description: An AWS Serverless Specification template describing your function.
Resources:
```



```
searchgateway:
  Type: 'AWS::Serverless::Function'
  Properties:
    FunctionName: gl-pdf-search-search-function-lambda
    Handler: lambda_function.lambda_handler
    Runtime: python3.9
    CodeUri: .
    Description: ''
    MemorySize: 128
    Timeout: 300
    Role: 'arn:aws:iam::032822161467:role/lambda-multirole'
    Layers:
      - 'arn:aws:lambda:us-east-1:032822161467:layer:pdf-search-aws-auth:1'
    VpcConfig:
      SecurityGroupIds:
        - 'sg-0ac3ef597cdb9b3a6'
        - 'sg-e7437fb3'
      SubnetIds:
        - 'subnet-a8eef986'
        - 'subnet-740dd539'
        - 'subnet-13f4e14f'
```

The repository and code pipeline was created in the same manner as the previous three lambda functions. Everything is identical except for the resource names. The screen shots have been omitted for brevity.

The resources created in the CloudFormation stack

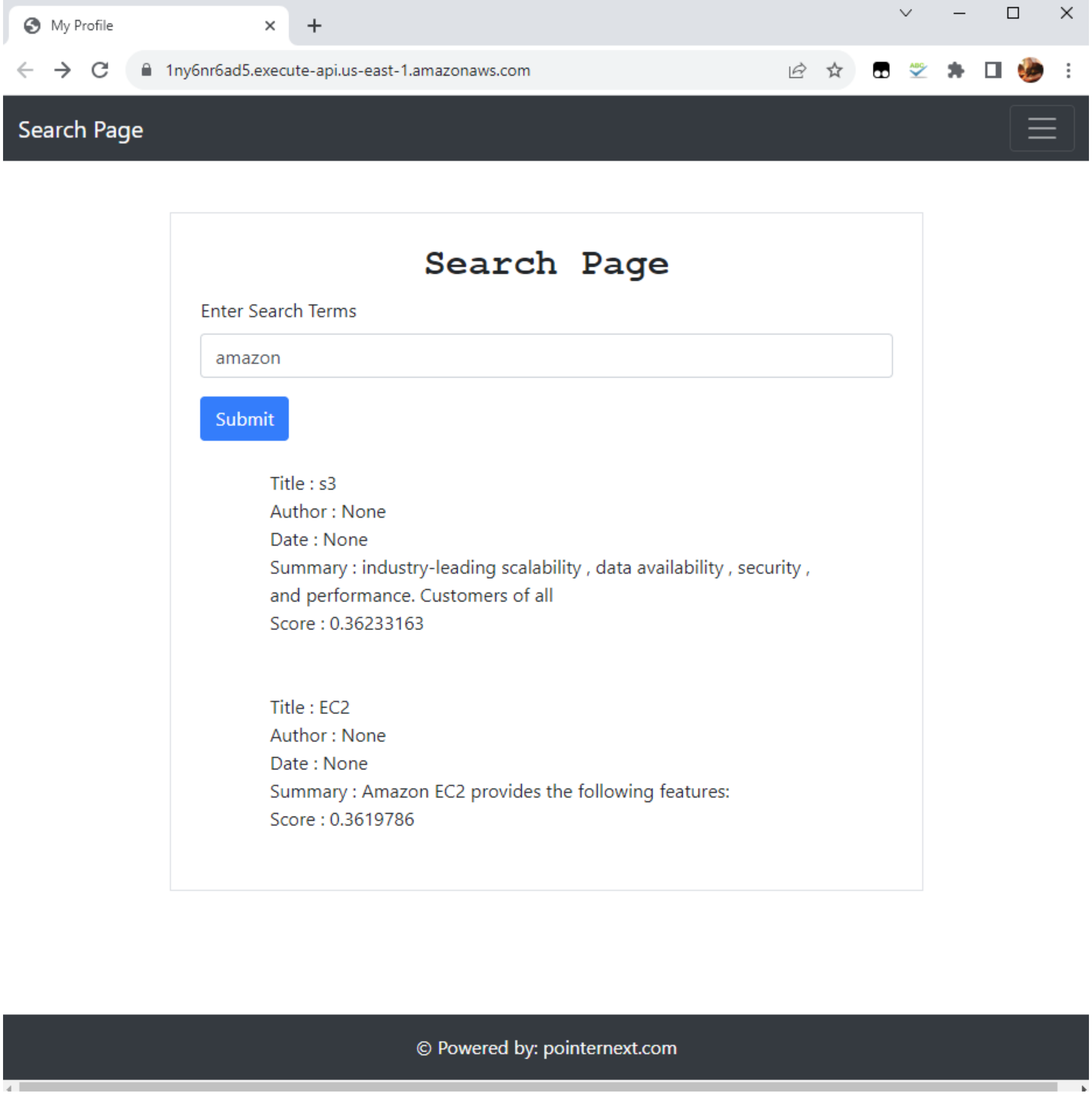


gl-pdf-search-search-gateway-stack

Resources (1)

Logical ID ▲	Physical ID ▽	Type ▽	Status ▽
searchgateway	gl-pdf-search-search-gateway-lambda	AWS::Lambda::Function	✓ UPDATE_COMPLETE

API HTTP Gateway: The API Gateway routes HTTP traffic to the lambda functions. The root route "/" is handled by SearchGateway lambda function to serve the "Search Page". The route "/search" is handled when an HTTP Post is submitted from the search page to serve the results.



Search Page

Enter Search Terms

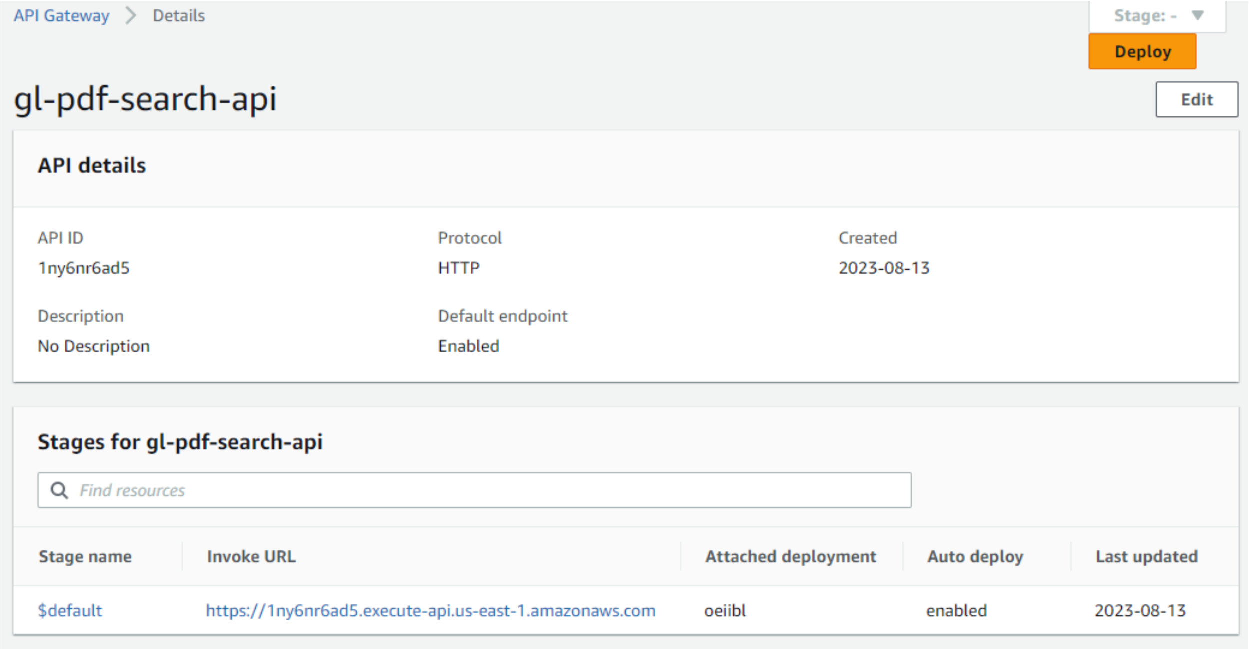
amazon

Title : s3
Author : None
Date : None
Summary : industry-leading scalability , data availability , security , and performance. Customers of all
Score : 0.36233163

Title : EC2
Author : None
Date : None
Summary : Amazon EC2 provides the following features:
Score : 0.3619786

© Powered by: pointernext.com

gl-pdf-search-api definition with the invoke URL.



gl-pdf-search-api

API details

API ID
1ny6nr6ad5

Protocol
HTTP

Created
2023-08-13

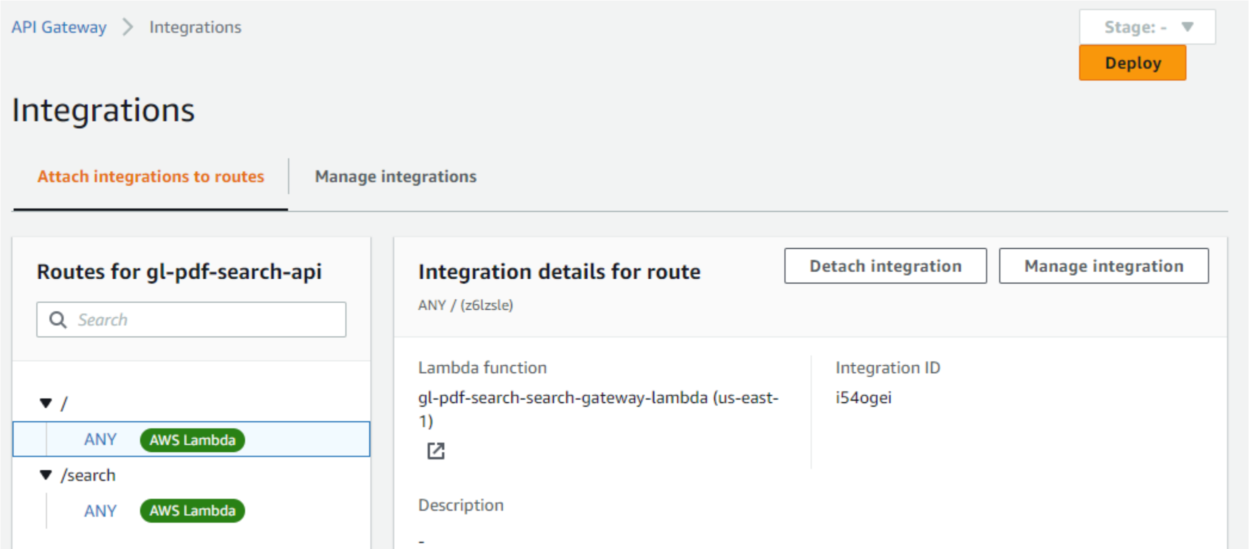
Description
No Description

Default endpoint
Enabled

Stages for gl-pdf-search-api

Stage name	Invoke URL	Attached deployment	Auto deploy	Last updated
$default	https://1ny6nr6ad5.execute-api.us-east-1.amazonaws.com	oeiibl	enabled	2023-08-13

Integration for route “/”. Handles any HTTP call. Probably should be simplified to just handle GET http calls.



Integrations

Routes for gl-pdf-search-api

- /
  - ANY AWS Lambda
- /search
  - ANY AWS Lambda

Integration details for route

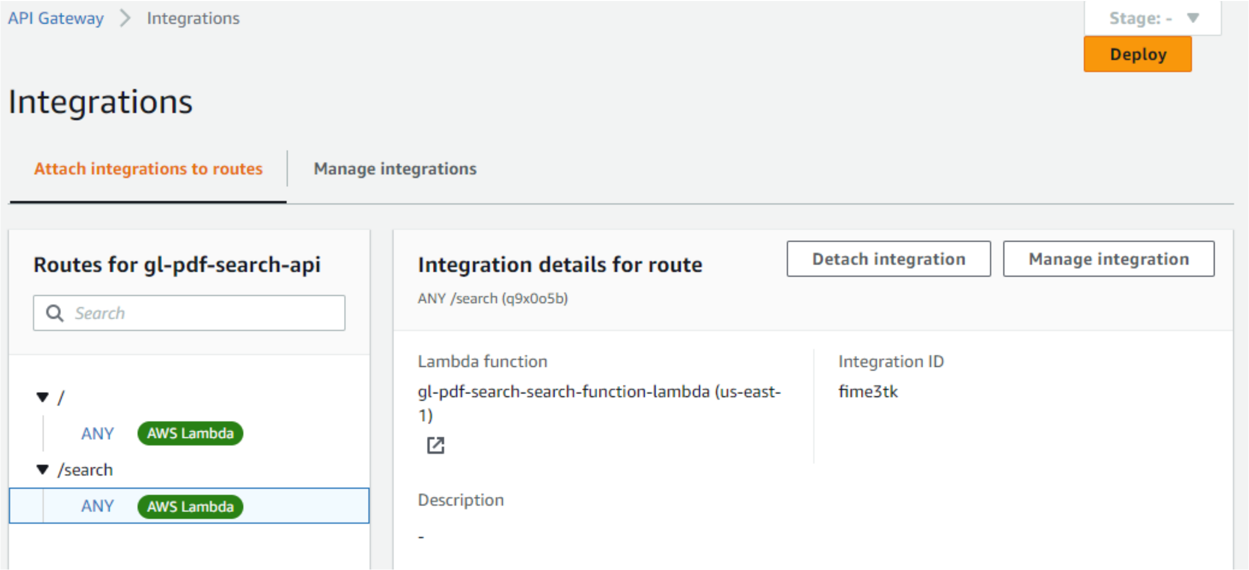
ANY / (z6lzsle)

Lambda function
gl-pdf-search-search-gateway-lambda (us-east-1)

Integration ID
i54ogei

Description
-

Integration for route “/search”. Handles any HTTP call. An alternative method would be to handle POST at “/” route. No need for “/search”.



Integrations

Routes for gl-pdf-search-api

- /
  - ANY AWS Lambda
- /search
  - ANY AWS Lambda

Integration details for route

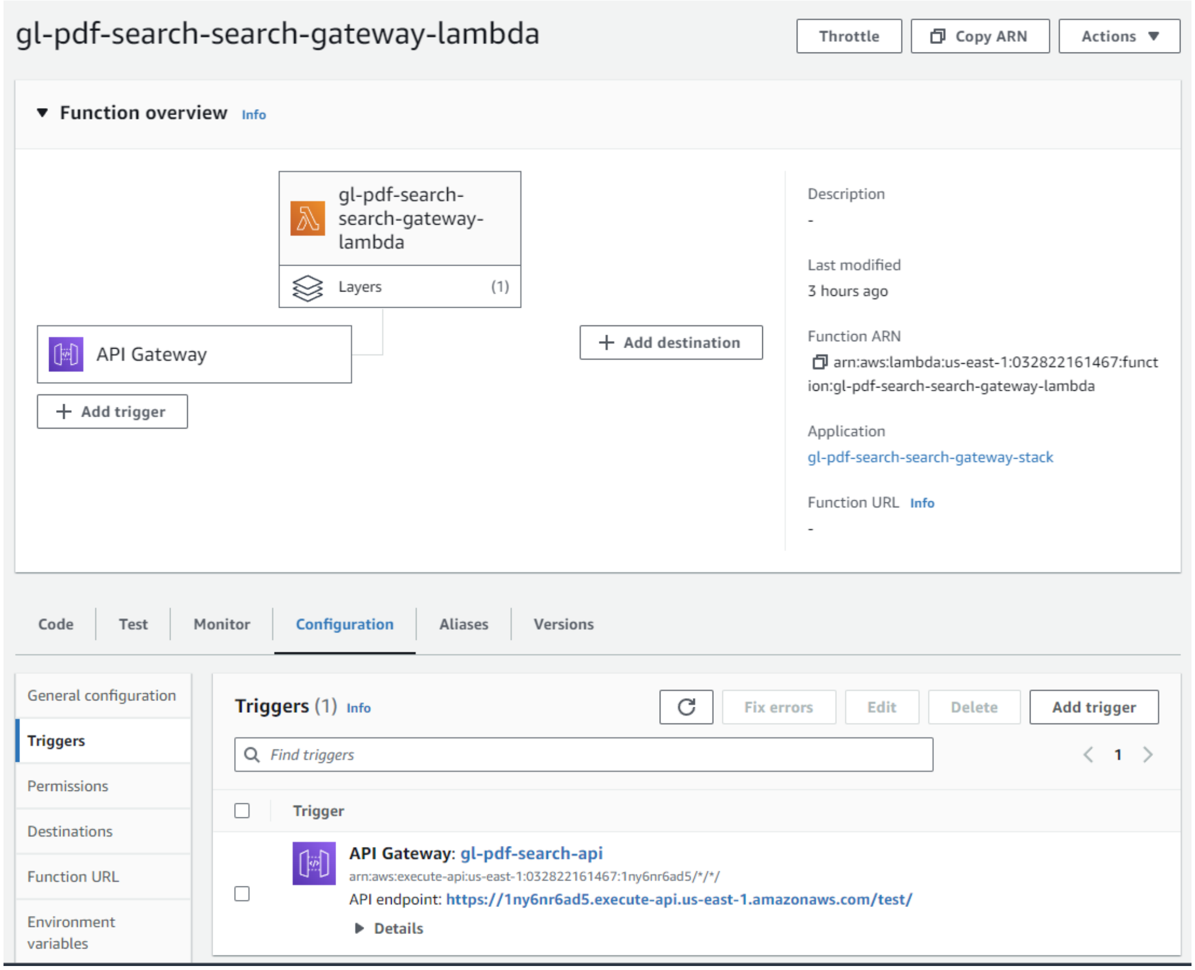
ANY /search (q9x0o5b)

Lambda function
gl-pdf-search-search-function-lambda (us-east-1)

Integration ID
fime3tk

Description
-

SearchGateway lambda function with the API Gateway trigger



gl-pdf-search-search-gateway-lambda

Description
-

Last modified
3 hours ago

Function ARN
arn:aws:lambda:us-east-1:032822161467:function:gl-pdf-search-search-gateway-lambda

Application
gl-pdf-search-search-gateway-stack

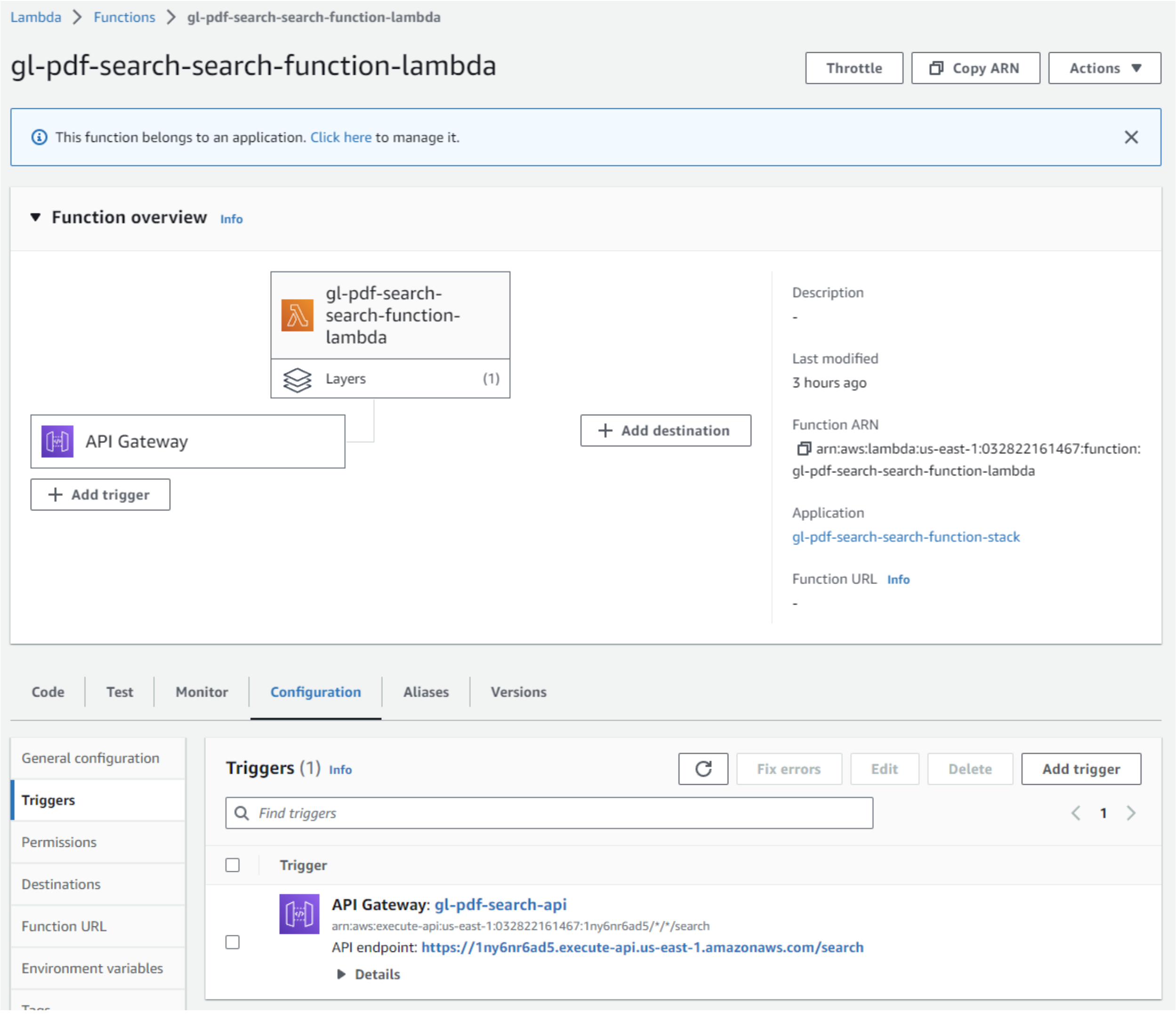
Function URL Info
-

Triggers (1) Info

□ Trigger

□ API Gateway: gl-pdf-search-api
arn:aws:execute-api:us-east-1:032822161467:1ny6nr6ad5/*/*/
API endpoint: https://1ny6nr6ad5.execute-api.us-east-1.amazonaws.com/test/

SearchFunction lambda function with the API Gateway trigger



gl-pdf-search-search-function-lambda

This function belongs to an application. Click here to manage it.

gl-pdf-search-search-function-lambda

Description
-

Last modified
3 hours ago

Function ARN
arn:aws:lambda:us-east-1:032822161467:function:gl-pdf-search-search-function-lambda

Application
gl-pdf-search-search-function-stack

Function URL
-

Triggers (1)

□	Trigger
□	API Gateway: gl-pdf-search-api arn:aws:execute-api:us-east-1:032822161467:1ny6nr6ad5/*/*/search API endpoint: https://1ny6nr6ad5.execute-api.us-east-1.amazonaws.com/search ▶ Details

Lessons & Observations:

This is one approach that could be taken to lower costs. The lambda functions are only billed by number of requests and execution times. This may be a little slow at times, especially if the lambda function has to be reloaded. But Concurrency Scaling and Availability Scaling can be used to improve performance.

It would probably make sense to keep the PDFToText and UploadToSearch lambda functions as they are as they are less likely to be called frequently and can run asynchronously. So latency isn't an issue.

A monolithic solution using EC2 Instances or perhaps one or more docker images hosted in EKS can be used for the front end piece to make them more responsive. AutoScaling and LoadBalancing can be used to improve performance.

Security isn't really taken into account when implementing this project. If this was a real production environment I would setup a dedicated VPC with private subnets and more restrictive security groups and IAM roles would be used.

cleanup.sh:

```bash
#!/bin/bash

# Delete HTTP API Gateway
apiId=$(aws apigatewayv2 get-apis --query "Items[?starts_with(Name, 'gl-pdf-search')].ApiId" --output text)
aws apigatewayv2 delete-api --api-id $apiId

# Empty S3 buckets
BUCKETS=$(aws s3api list-buckets --query "Buckets[?starts_with(Name, 'gl-pdf-search')].Name" --output text)
for BUCKET in $BUCKETS; do
    aws s3 rm s3://$BUCKET --recursive
done

# Delete CloudFormation stacks
STACKS=$(aws cloudformation list-stacks --query "StackSummaries[?starts_with(StackName, 'gl-pdf-search') && StackStatus != 'DELETE_COMPLETE'].StackName" --output text)
for STACK in $STACKS; do
    aws cloudformation delete-stack --stack-name $STACK
done

# Delete CodePipelines
PIPELINES=$(aws codepipeline list-pipelines --query "pipelines[?starts_with(name, 'gl-pdf-search')].name" --output text)
for PIPELINE in $PIPELINES; do
    aws codepipeline delete-pipeline --name $PIPELINE
done

# Delete CodeBuilds
PROJECTS=$(aws codebuild list-projects --query "projects[?starts_with(@, 'gl-pdf-search')]" --output text)
for PROJECT in $PROJECTS; do
    aws codebuild delete-project --name $PROJECT
done

# Delete CodeCommit repositories
```

```
REPOSITORIES=$(aws codecommit list-repositories --query
"repositories[?starts_with(repositoryName, 'gl-pdf-search')].repositoryName" --
output text)
for REPOSITORY in $REPOSITORIES; do
    aws codecommit delete-repository --repository-name $REPOSITORY
done

# Delete Lambda layers
LAYERS=$(aws lambda list-layers --query "Layers[?starts_with(LayerName, 'pdf-
search')].LayerName" --output text)
for LAYER in $LAYERS; do
    VERSIONS=$(aws lambda list-layer-versions --layer-name $LAYER --query
"LayerVersions[].Version" --output text)
    for VERSION in $VERSIONS; do
        aws lambda delete-layer-version --layer-name $LAYER --version-number
$VERSION
    done
done

# Force delete S3 buckets
BUCKETS=$(aws s3api list-buckets --query "Buckets[?starts_with(Name, 'gl-pdf-
search')].Name" --output text)
for BUCKET in $BUCKETS; do
    aws s3 rb s3://$BUCKET --force
done

# Delete OpenSearch Service domain
aws opensearch delete-domain --domain-name pdf-search

# Stop bastion instance
INSTANCE_ID=$(aws ec2 describe-instances --filters "Name=tag:Name,Values=user-
mgmt-portal-bastion" --query "Reservations[].Instances[].InstanceId" --output
text)
aws ec2 stop-instances --instance-ids $INSTANCE_ID
```