Primena Serverless arhitekture uz AWS Lambda, SQS i S3 servise

Veljko Veljović 1937

Napredno Softversko inženjerstvo

Sadržaj

Zašto AWS?

U savremenim distribuiranim aplikacijama, skalabilnost, dostupnost i sigurnost predstavljaju ključne izazove. Tradicionalni lokalni serveri i manuelno upravljanje infrastrukturom često dovode do problema kao što su:

- Smanjena dostupnost aplikacije u slučaju hardverskih problema
- Teškoće u skaliranju aplikacije prema potražnji korisnika
- Nedostatak automatizacije u postavljanju i održavanju servisa

Rešenje uz pomoć AWS-a

Rešenje uz pomoć AWS servisa

Platforma Amazon Web Services (AWS) omogućava efikasno rešavanje izazova modernih aplikacija kao što su skalabilnost, dostupnost i bezbednost. Umesto manuelnog upravljanja infrastrukturom, AWS pruža skup servisa koji omogućavaju automatizaciju i sigurnost sistema.

AWS nudi sledeće prednosti:

- Omogućava skalabilno i dostupno izvršavanje servisa uz pomoć servisa kao što su EC2, Lambda i Elastic Load Balancing, koji automatski prilagođavaju resurse u skladu sa opterećenjem.
- Omogućava čuvanje i serviranje statičkih fajlova, slika i drugih resursa korišćenjem S3 bucket-a.
- Omogućava asinhronu komunikaciju između komponenti sistema pomoću Amazon SQS (Simple Queue Service), čime se povećava otpornost i skalabilnost distribuiranih aplikacija.

AWS SQS

Amazon SQS (Simple Queue Service) je u potpunosti upravljani servis za razmenu poruka u redovima, koji omogućava asinhronu komunikaciju između komponenti distribuiranog sistema.

- Asinhrona obrada
- Skalabilnost i otpornost
- Decoupling servisa
- DLQ (Dead Letter Queue)

Zašto koristiti SQS?

Amazon SQS povećava pouzdanost, skalabilnost i fleksibilnost aplikacije. Uklanja uska grla u komunikaciji između mikroservisa i omogućava stabilan rad sistema i pod opterećenjem.

AWS SQS

AWS Lambda

AWS Lambda – Serverless izvršavanje koda

- Potpuno serverless
- Event-driven model
- Automatsko skaliranje
- Plaćanje po pozivu
- Jednostavna integracija sa ostalim AWS servisima

Kada koristiti AWS Lambde?

- Obrada događaja
- Automatizacija zadataka
- Mikroservisna arhitektura

AWS S3

Amazon S3 (Simple Storage Service) je servis za čuvanje objekata koji omogućava bezbedno skladištenje i pristup bilo kojoj količini podataka – u bilo kom trenutku i sa bilo kog mesta.

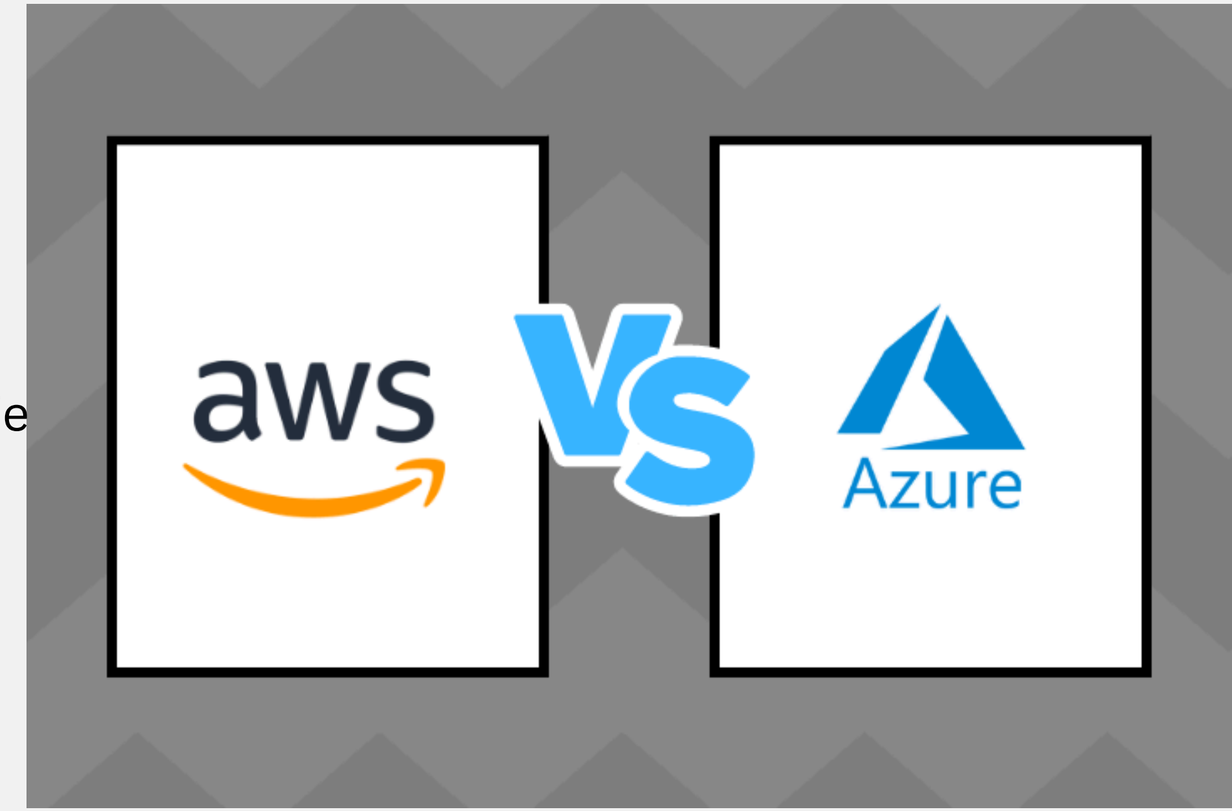
S3 je idealan za:
Statičke fajlove aplikacije (slike, dokumenti, video fajlovi)
Backup i arhiviranje podataka
Logove i analitičke podatke
Hosting statičkih web sajtova

Karakteristike:

- Visoka dostupnost i pouzdanost
- Skalabilnost
- Integracija sa drugim AWS servisima
- Plaćanje po korišćenju

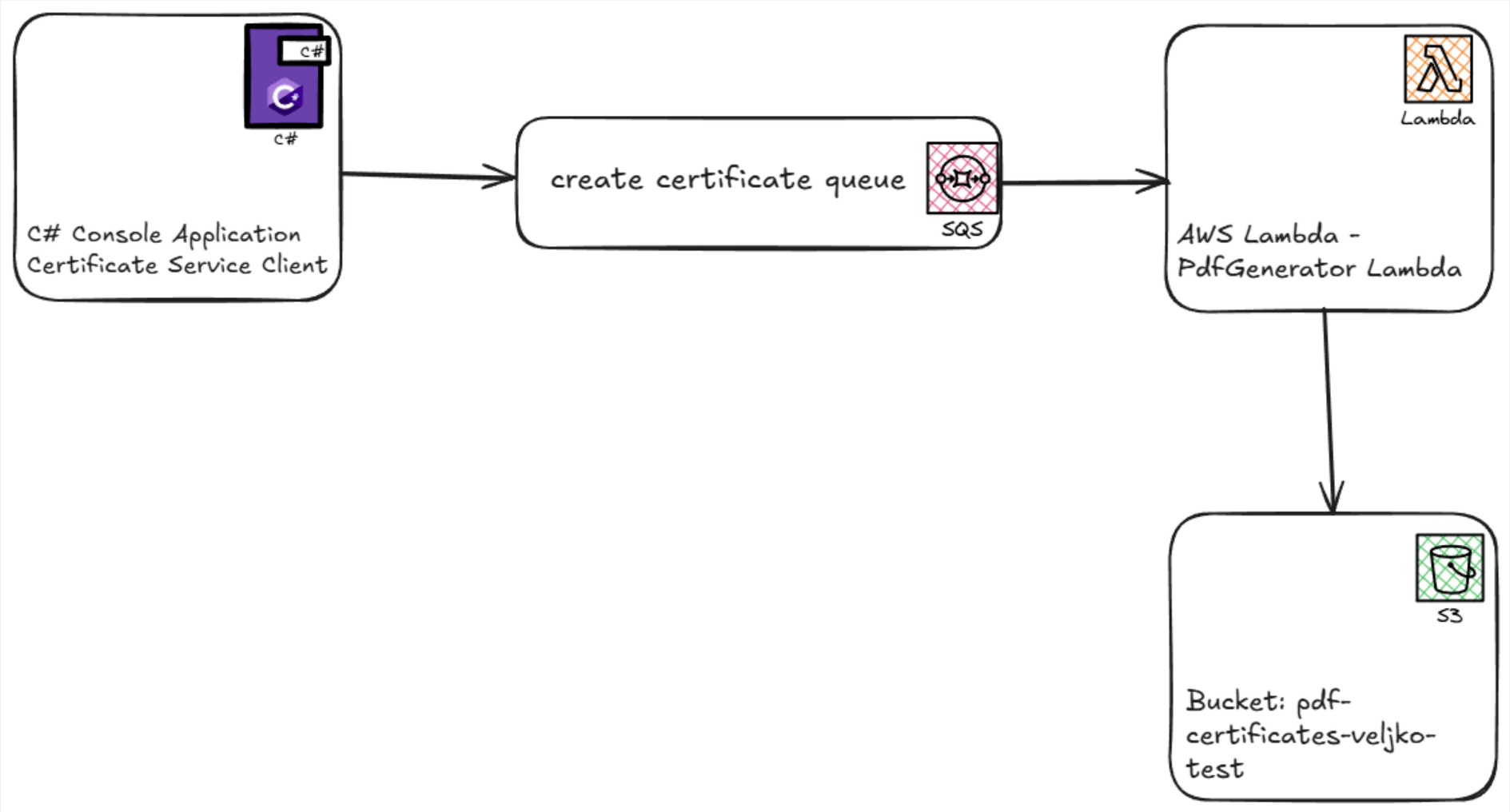
AWS vs Azure

- Azure se bolje integriše sa Microsoft proizvodima, dok je AWS platformno neutralan.
- AWS ima složeniji cenovni model, dok je Azure jednostavniji, naročito za Windows okruženja.
- AI i mašinsko učenje: AWS SageMaker je moćniji za ML
- AWS nudi kvalitetnijemrežne servise sa većom fleksibilnošću i nižom latencijom.
- Free tier i učenje: AWS ima opširniji free tier i lakši je za početnike.



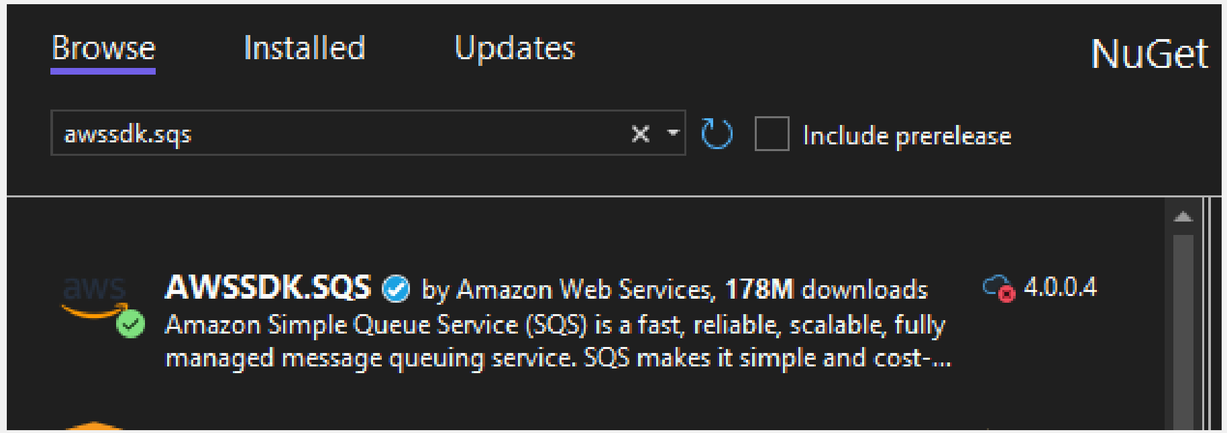
Implementacija Projekta

Dijagram sistema



Certificate Service Client

```
while (true)
{
    Console.WriteLine("Enter candidate's first name");
    var name = Console.ReadLine();

    Console.WriteLine("Enter candidate's last name");
    var lastName =  Console.ReadLine();

    Console.WriteLine("Enter course name");
    var courseName = Console.ReadLine();

    var secrets = new List<string>();
    if(secrets.Count > 0)
    {
        secrets = File.ReadLines(Constants.SecretPath)
        .ToList();
    }

    var pubslichCert = new PublishCertificate(new AmazonSQSClient(secrets[0],
        secrets[1],
        RegionEndpoint.USEast1));

    var model = new CertificatesModel(name, lastName, courseName);

    await pubslichCert.Publish(model);
}
```




```
namespace CertificateServiceClient
{
    1 reference
    public sealed class PublishCertificate(IAmazonSQS sqsClient) : IPublishCertifica
    {
        2 references
        public async Task Publish(CertificatesModel certificate)
        {
            var serializedCertificate = JsonSerializer.Serialize(certificate);

            var messageRequest = new SendMessageRequest()
            {
                QueueUrl = Constants.QueueUrl,
                MessageBody = serializedCertificate
            };

            await sqsClient.SendMessageAsync(messageRequest);
        }
    }
}
```

SQS

Create queue

Details

Type

Choose the queue type for your application or cloud infrastructure.

- (•) **Standard** Info
  At-least-once delivery, message ordering isn't preserved
  - At-least once delivery
  - Best-effort ordering
- ( ) **FIFO** Info
  First-in-first-out delivery, message ordering is preserved
  - First-in-first-out delivery
  - Exactly-once processing

ⓘ You can't change the queue type after you create a queue.

Name

create-certificate-queue

A queue name is case-sensitive and can have up to 80 characters. You can use alphanumeric characters, hyphens (-), and underscores (_).

Configuration Info

Set the maximum message size, visibility to other consumers, and message retention.

Visibility timeout | Info

30 Seconds

Should be between 0 seconds and 12 hours.

Message retention period | Info

4 Days

Should be between 1 minute and 14 days.

Delivery delay | Info

0 Seconds

Should be between 0 seconds and 15 minutes.

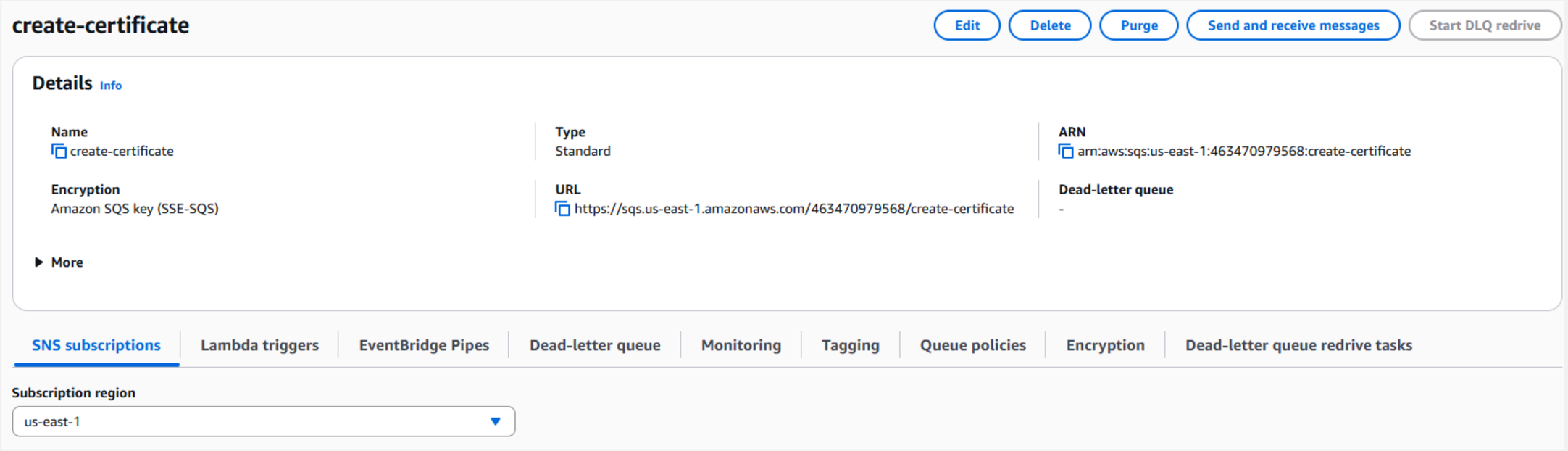
Maximum message size | Info

256 KB

Should be between 1 KB and 256 KB.

Receive message wait time | Info

SQS


create-certificate
Edit
Delete
Purge
Send and receive messages
Start DLQ redrive
Details Info
Name
create-certificate
Type
Standard
ARN
arn:aws:sqs:us-east-1:463470979568:create-certificate
Encryption
Amazon SQS key (SSE-SQS)
URL
https://sqs.us-east-1.amazonaws.com/463470979568/create-certificate
Dead-letter queue
-
More
SNS subscriptions
Lambda triggers
EventBridge Pipes
Dead-letter queue
Monitoring
Tagging
Queue policies
Encryption
Dead-letter queue redrive tasks
Subscription region
us-east-1

PdfGeneratorLambda

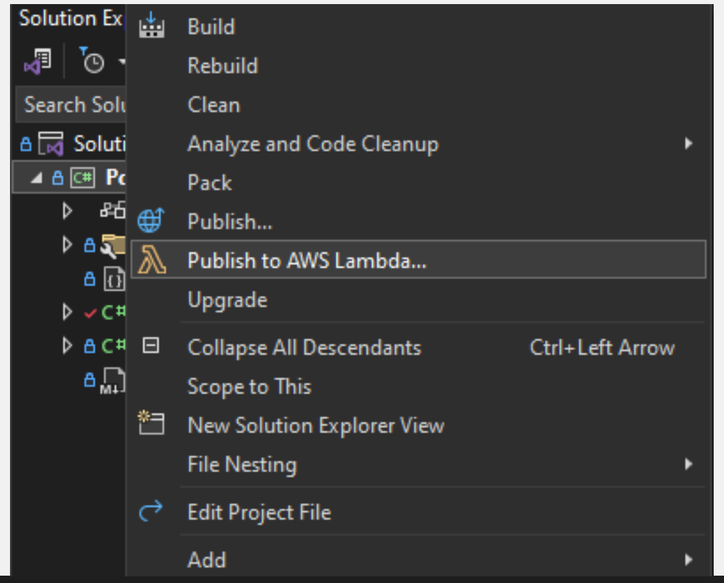
```
<ItemGroup>
  <PackageReference Include="Amazon.Lambda.Core" Version="2.5.0" />
  <PackageReference Include="Amazon.Lambda.Serialization.SystemTextJson" Version="2.4.4" />
  <PackageReference Include="Amazon.Lambda.SQSEvents" Version="2.2.0" />
  <PackageReference Include="AWSSDK.S3" Version="4.0.0.7" />
  <PackageReference Include="QuestPDF" Version="2025.5.0" />
</ItemGroup>
```

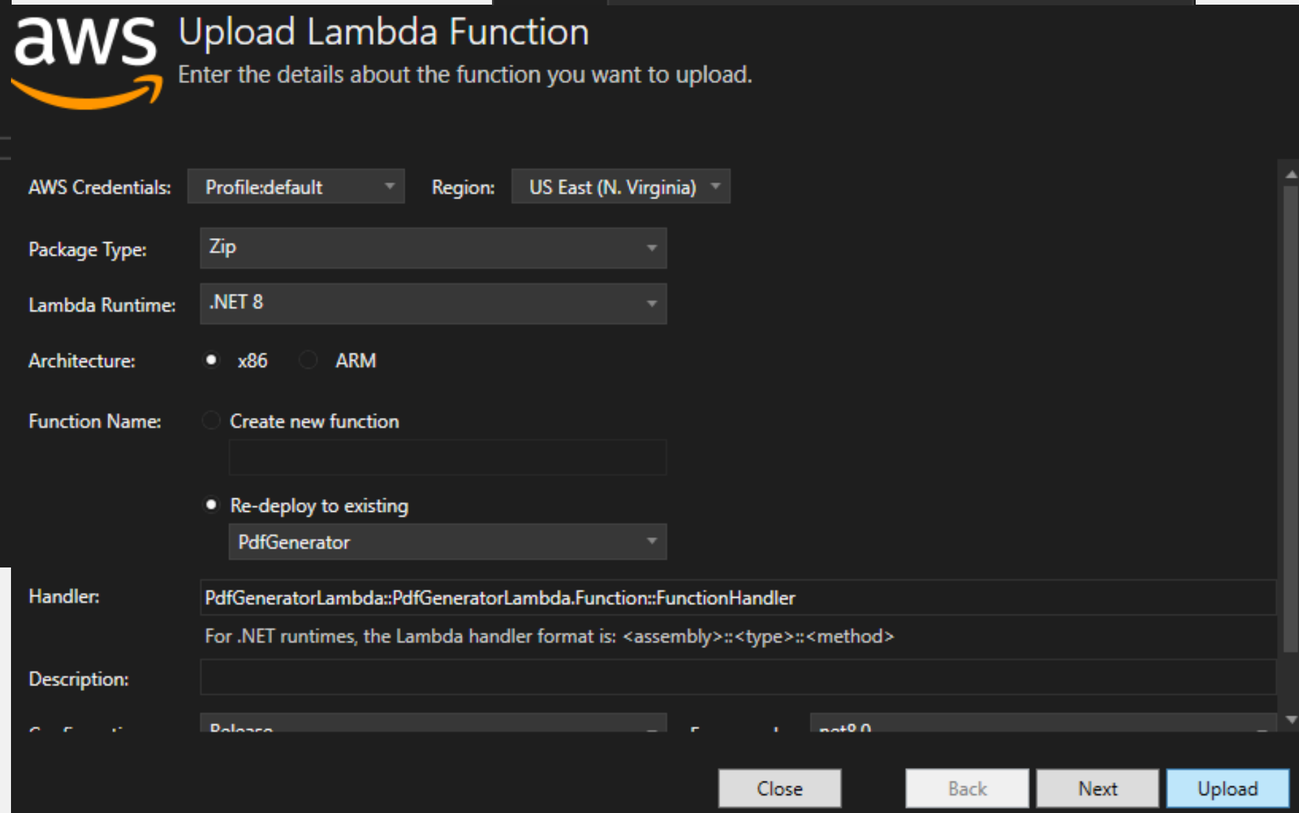
```
private readonly IAmazonS3 s3Client;
0 references
public Function()
{
    s3Client = new AmazonS3Client();
}

0 references
public async Task FunctionHandler(SQSEvent evnt, ILambdaContext context)
{
    foreach(var message in evnt.Records)
    {
        await ProcessMessageAsync(message, context);
    }
}
```

```
1 reference
private async Task ProcessMessageAsync(SQSEvent.SQSMessage message, ILambdaContext context)
{
    context.Logger.LogInformation($"Processed message {message.Body}");

    var certificate = JsonSerializer.Deserialize<CertificatesModel>(message.Body);
    using var stream = GeneratePdfInMemory(certificate);

    var request = new PutObjectRequest
    {
        BucketName = "pdf-certificates-veljko-test",
        Key = $"{certificate.FirstName}-{certificate.LastName}-{certificate.CourseName}",
        InputStream = stream,
        ContentType = "application/pdf"
    };
    await s3Client.PutObjectAsync(request);
}
```

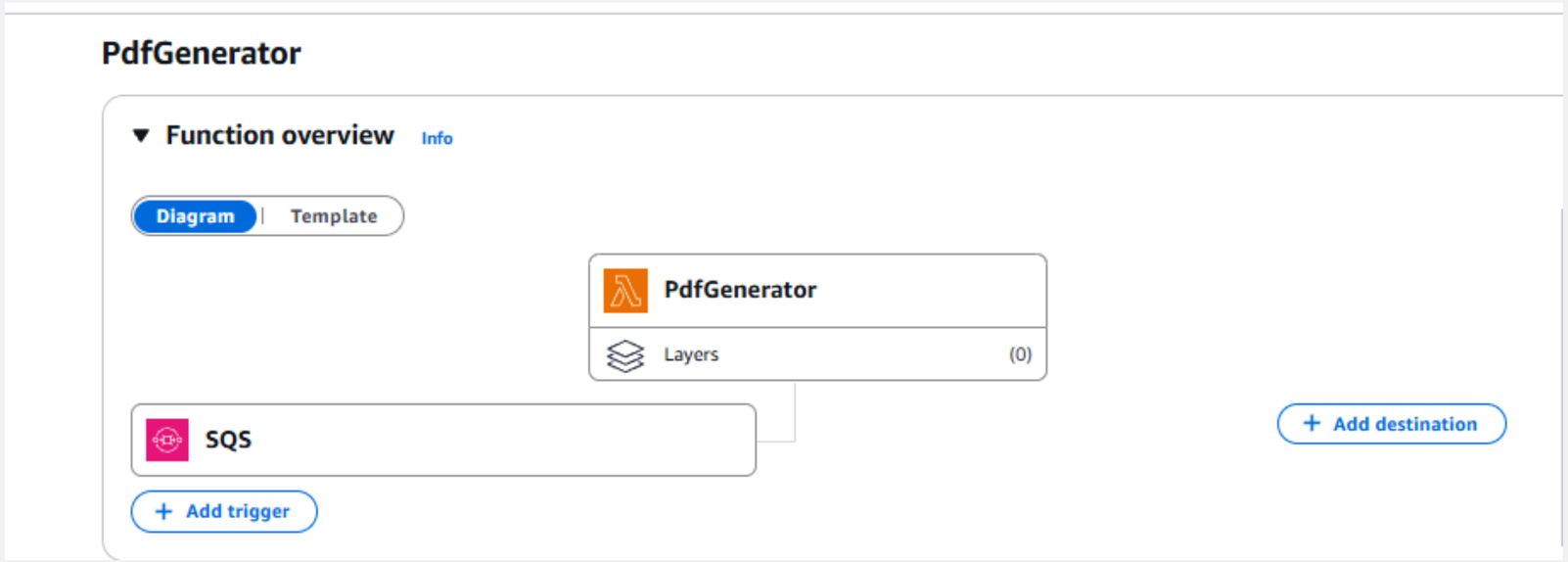
PdfGeneratorLambda

```
1 reference
public static MemoryStream GeneratePdfInMemory(CertificatesModel model)
{
    QuestPDF.Settings.License = LicenseType.Community;

    var pdfBytes =  Document.Create(container =>
    {
        container.Page(page =>
        {
            page.Margin(50);
            page.Size(PageSizes.A4);
            page.PageColor(Colors.White);
            page.DefaultTextStyle(x => x.FontSize(20).FontFamily("Times New Roman"));

            page.Header().Text("Certificate of Completion")
                .FontSize(36)
                .Bold()
                .FontColor(Colors.Blue.Medium)
                .AlignCenter();

            page.Content().PaddingVertical(50).Column(column =>
            {
                column.Item().AlignCenter().Text("This certificate is proudly presented to").FontSize(20);
                column.Item().PaddingVertical(15).AlignCenter().Text(model.FirstName).FontSize(30).Bold();
                column.Item().AlignCenter().Text("for successfully completing the course:").FontSize(20);
                column.Item().PaddingBottom(10).AlignCenter().Text(model.CourseName).FontSize(26).Bold();
                column.Item().AlignCenter().Text($"Date: {DateTime.Today:MMMM dd, yyyy}").FontSize(16);
                column.Item().PaddingTop(50).AlignRight().Text("____________________").FontSize(20);
                column.Item().AlignRight().Text("Instructor's Signature").FontSize(16);
            });

            page.Footer().AlignCenter().Text("Congratulations!").FontSize(18).Italic().FontColor(Colors.Green.Darken1);
        });
    }).GeneratePdf();

    return new MemoryStream(pdfBytes);
}
```

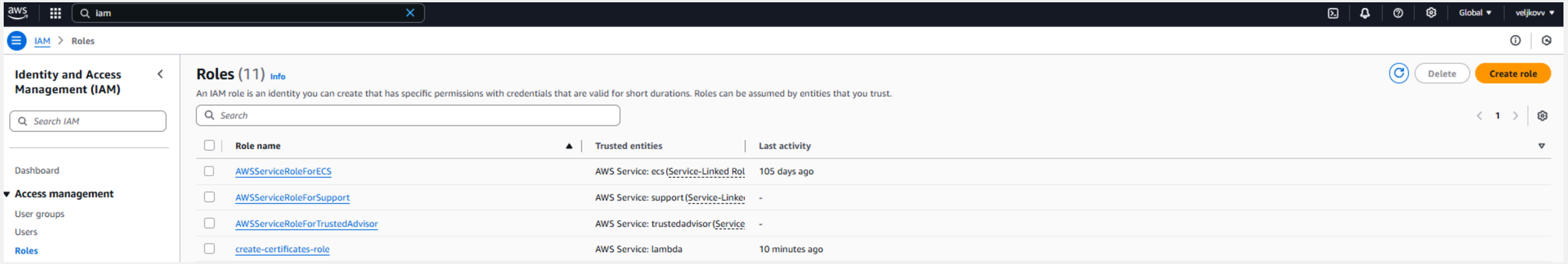


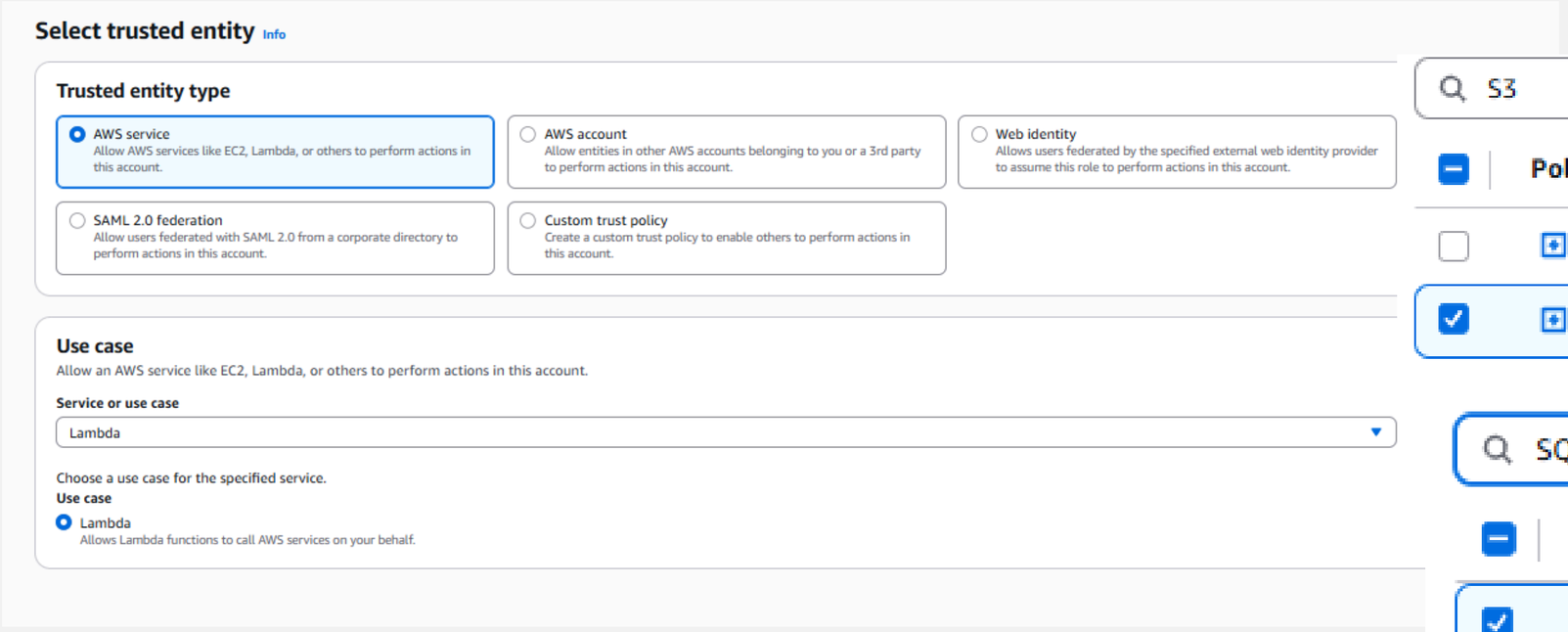


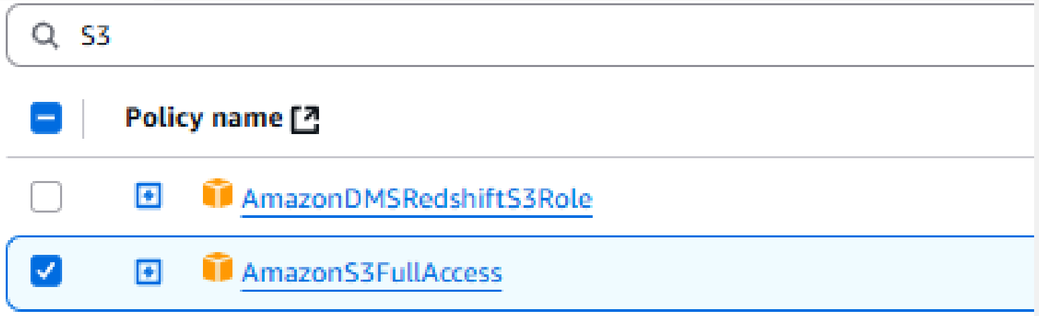
Lambda

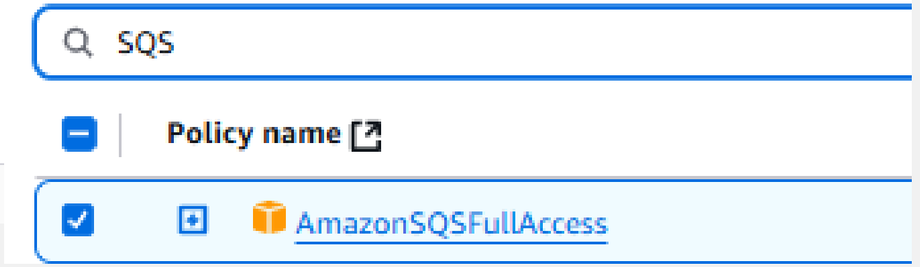

PdfGenerator
Function overview
Info
Diagram
Template
PdfGenerator
Layers
(0)
SQS
Add destination
Add trigger

IAM


aws
Search [Alt+S]
Global
velikovv
IAM > Roles
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Roles (11) Info
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.
Delete
Create role
Search
1
Role name
Trusted entities
Last activity
AWSServiceRoleForECS
AWS Service: ecs (Service-Linked Rol
105 days ago
AWSServiceRoleForSupport
AWS Service: support (Service-Linke
-
AWSServiceRoleForTrustedAdvisor
AWS Service: trustedadvisor (Service
-
create-certificates-role
AWS Service: lambda
10 minutes ago


Select trusted entity Info
Trusted entity type
AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.
Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.
Service or use case
Lambda
Choose a use case for the specified service.
Use case
Lambda
Allows Lambda functions to call AWS services on your behalf.


S3
Policy name
AmazonDMSRedshiftS3Role
AmazonS3FullAccess


SQS
Policy name
AmazonSQSFullAccess

IAM

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

create-certificate

Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description

Code | Test | Monitor | Configuration | Aliases | Versions

- General configuration
- Triggers
- Permissions
- Destinations
- Function URL
- Environment variables
- Tags
- VPC
- RDS databases
- Monitoring and operations tools
- Concurrency and recursion detection
- Asynchronous invocation
- Code signing
- File systems
- State machines

Execution role

Edit | View role document

Role name
create-certificates-role

Resource summary

To view the resources and actions that your function has permission to access, choose a service.

Amazon S3
1 action, 1 resource

By action | By resource

Resource	Actions
All resources	Allow: s3:*

Lambda obtained this information from the following policy statements:

- Managed policy AmazonS3FullAccess, statement 0

Resource-based policy statements Info

View policy | Edit | Delete | Add permissions

Resource-based policies grant other AWS accounts and services permissions to access your Lambda resources.

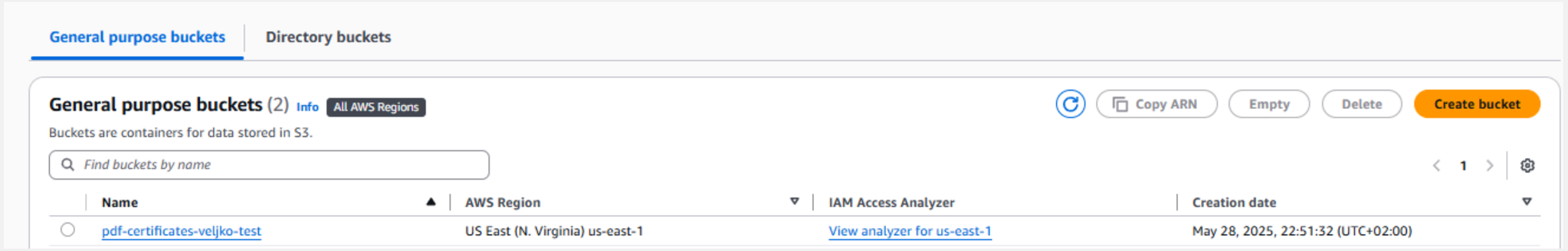
Find policy statements

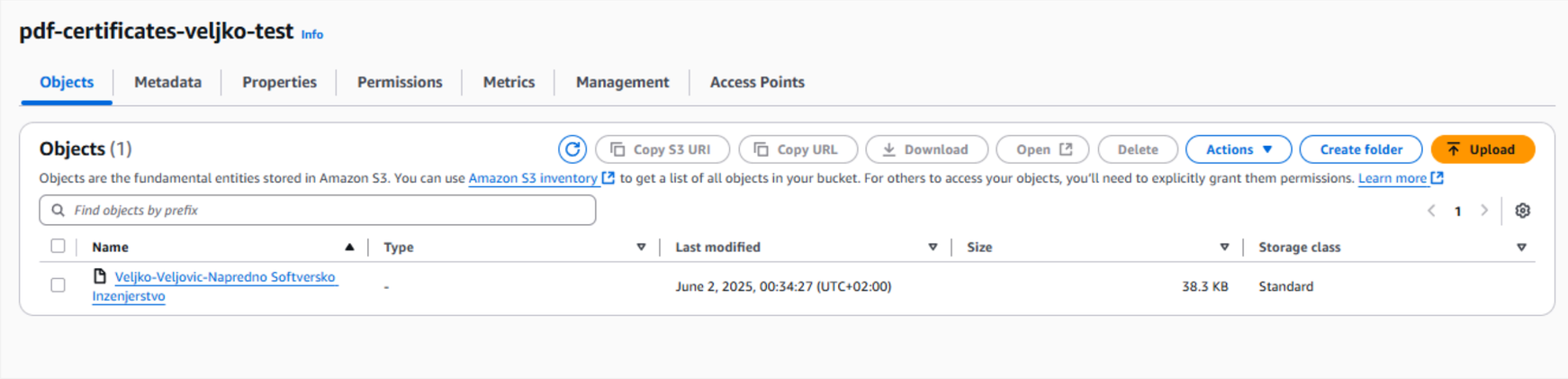
1

Statement ID	Principal	PrincipalOrgID	Conditions	Actio

No policy statements

S3


General purpose buckets
Directory buckets
General purpose buckets (2) Info All AWS Regions
Buckets are containers for data stored in S3.
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
1
Name
AWS Region
IAM Access Analyzer
Creation date
pdf-certificates-veljko-test
US East (N. Virginia) us-east-1
View analyzer for us-east-1
May 28, 2025, 22:51:32 (UTC+02:00)


pdf-certificates-veljko-test Info
Objects
Metadata
Properties
Permissions
Metrics
Management
Access Points
Objects (1)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
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
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
Veljko-Veljovic-Napredno Softversko Inzenjerstvo
-
June 2, 2025, 00:34:27 (UTC+02:00)
38.3 KB
Standard

S3

Certificate of Completion

This certificate is proudly presented to

Veljko Veljovic

for successfully completing the course:

Napredno Softversko Inzenjerstvo

Date: June 01, 2025

_____

Instructor's Signature

Hvala na pažnji!