

Microsoft .NET on the AWS Cloud: looking over the hedge

BY ROB VAN PAMEL



WHY ?



A little story

- .NET developer since 2007
- Preference for C# development

I was an Azure fanboy ... ;-)

".NET development == Microsoft knows best ..."

- 2015: Build a next generation

Greenfield project

High competition

Cost optimization

How did AWS convinced me . . . ?

AXXES

Why AWS?

- Large community of customers
 - Netflix
 - Facebook
 - Nasa
 - BBC
- Broad set of products
 - Compute
 - Storage
 - Database
 - Mobile
 - IoT
 - ...

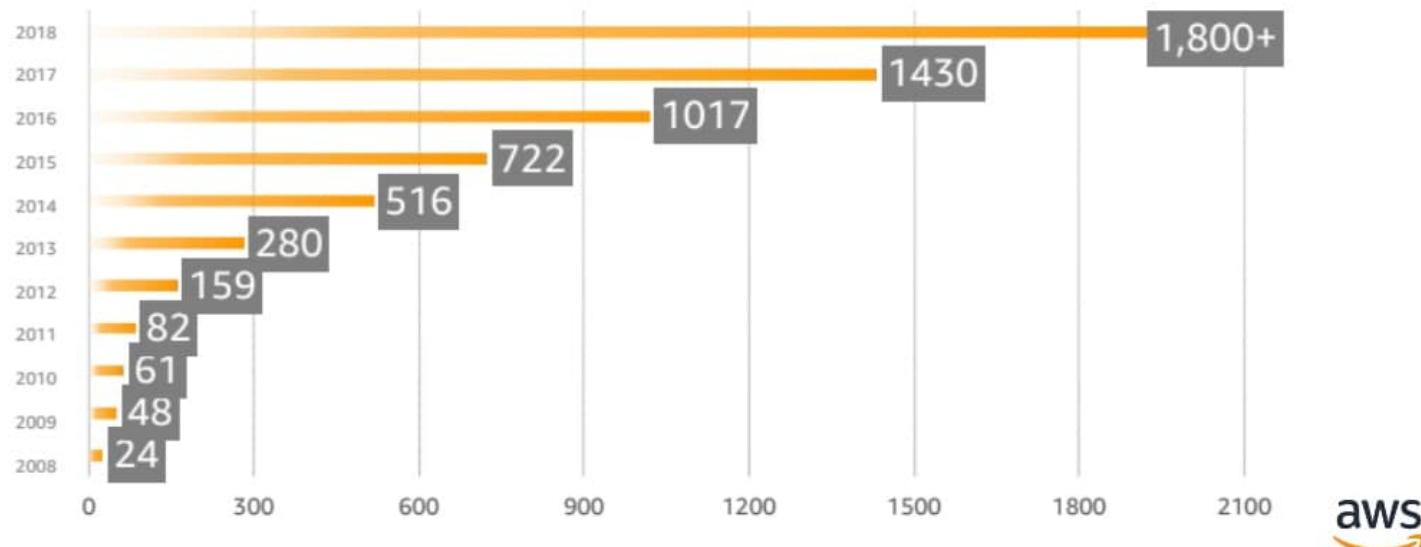


Mindset

- Pace of innovation
- Open mind
 - Open source contributions
- Listen to customers
 - ECS vs EKS
 - Dynamodb and DocumentDb (Mongo)
 - Kinesis vs AWS MSK (Kafka)

AWS' Pace of Innovation

90%+ of those new features were a direct result of feedback from our customers and invent on behalf of customers



AXES

<https://dotnetfoundation.org/>

Long commitment to .NET
& Windows workloads

Corporate sponsor

<https://aws.amazon.com/developer/language/net/>

AXXES



COMMUNITY

- Very active & good support
 - Developer advocates
- OpenSource
 - <https://github.com/aws/dotnet>
 - <https://www.nuget.org/profiles/awsdotnet>
 - Battle tested
- User group: <https://www.meetup.com/AWS-User-Group-Belgium/>

'Amazon's investment in the .NET Foundation and support for .NET developers is broad-minded. Being able to take all of my skills and investment in .NET development and the related IDEs and use them on AWS is valuable.'

*-
Julie Lerman, .NET Community Leader*

Getting started

- Create Account
- Free Tier : <https://aws.amazon.com/free>



Always free

These free tier offers do not expire and are available to all AWS customers



12 months free

Enjoy these offers for 12-months following your initial sign-up date to AWS



Trials

Short-term free trial offers start from the date you activate a particular service



AWS Management Console

AWS services

▼ Recently visited services



RDS



EC2



Elastic Beanstalk



IAM

▼ All services



Compute

EC2

Lightsail

Lambda

Batch

Elastic Beanstalk

Serverless Application Repository

AWS Outposts

EC2 Image Builder

AWS App Runner



Containers

Elastic Container Registry

Elastic Container Service

Elastic Kubernetes Service

Red Hat OpenShift Service on AWS



Storage

S3

EFS

FSx



Developer Tools

CodeStar

CodeCommit

CodeArtifact

CodeBuild

CodeDeploy

CodePipeline

Cloud9

CloudShell

X-Ray

AWS FIS



Customer Enablement

AWS IQ

Support

Managed Services

Activate for Startups



Robotics

AWS RoboMaker



Blockchain



Machine Learning

Amazon SageMaker

Amazon Augmented AI

Amazon CodeGuru

Amazon DevOps Guru

Amazon Comprehend

Amazon Forecast

Amazon Fraud Detector

Amazon Kendra

Amazon Lex

Amazon Personalize

Amazon Polly

Amazon Rekognition

Amazon Textract

Amazon Transcribe

Amazon Translate

AWS DeepComposer

AWS DeepLens

AWS DeepRacer

AWS Panorama

Amazon Monitron



AWS Cost Management

AWS Cost Explorer

AWS Budgets

AWS Marketplace Subscriptions

AWS Application Cost Profiler



Front-end Web & Mobile

AWS Amplify

Mobile Hub

AWS AppSync

Device Farm

Amazon Location Service



AR & VR

Amazon Sumerian



Application Integration

Step Functions

Amazon AppFlow

Amazon EventBridge

Amazon MQ

Simple Notification Service

Stay connected to your AWS resources on-the-go



AWS Console Mobile App now supports four additional regions. Download the AWS Console Mobile App to your iOS or Android mobile device. [Learn more](#)

Explore AWS

Save money with Amazon Location Maps

Replace existing maps in your web application or add a new map to display store locations and other geo-tagged data cost effectively. [Learn more](#)

Amazon Location Service

Easily and securely add maps, search for points of interest, geocoding, routes, tracking, and geofencing to your application. [Get started](#)

Introducing Amazon FSx for NetApp ONTAP

Fully managed storage built on NetApp's popular ONTAP file system. [Learn more](#)

Build Apps Faster with GraphQL

AWS AppSync uses GraphQL APIs to query data from multiple data sources in a single request. [Get started](#)



Sample API

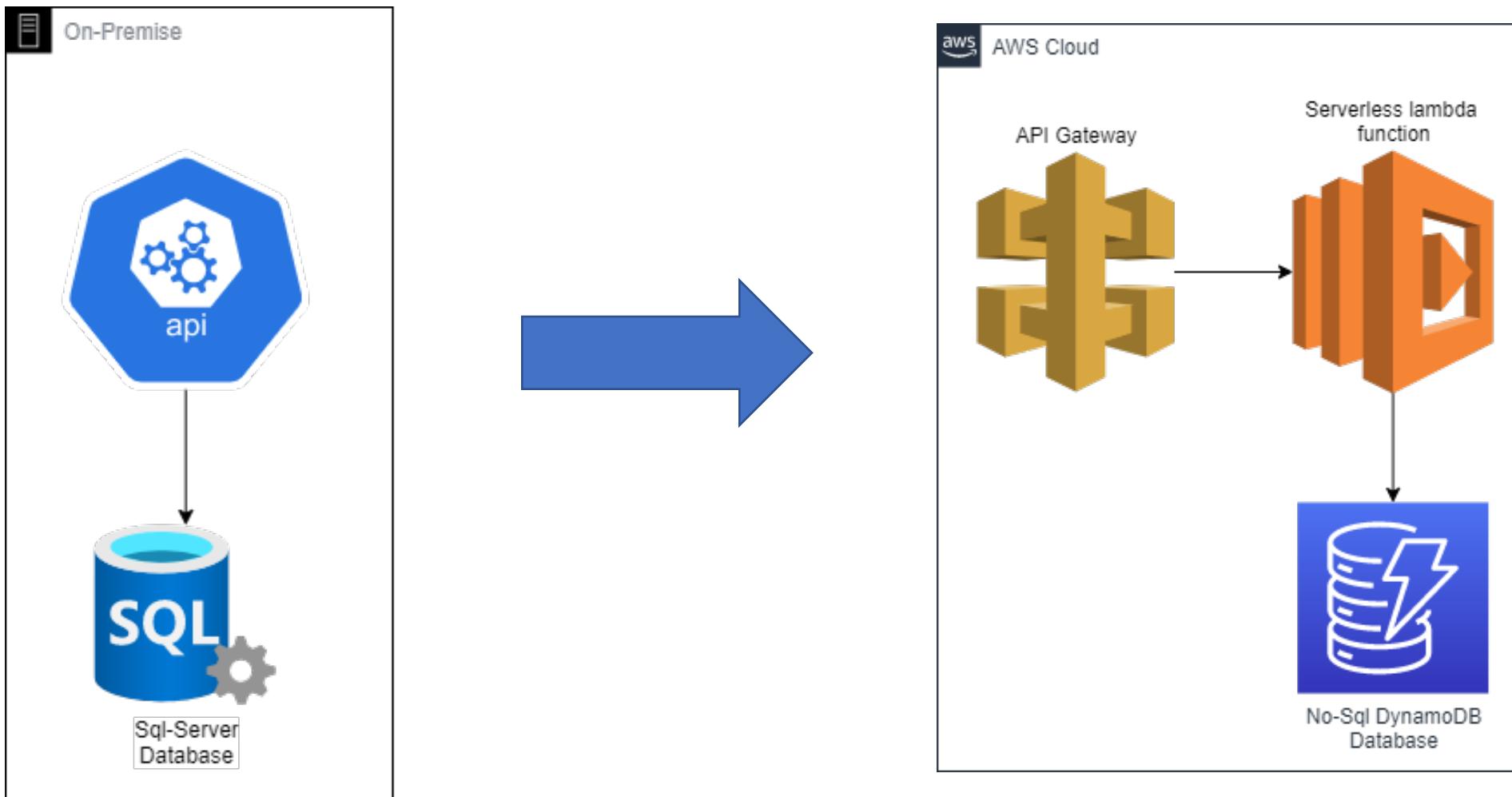
- Conference Application
- Speakers
- Talks

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the `Program.cs` file for the project `LookingOverTheHedge.Api`. The code implements a `Program` class with a `Main` method that creates a host builder and runs it. The `CreateHostBuilder` method uses `Host.CreateDefaultBuilder` and `ConfigureWebHostDefaults` to set up the web host. The `Startup` class is used for configuration.

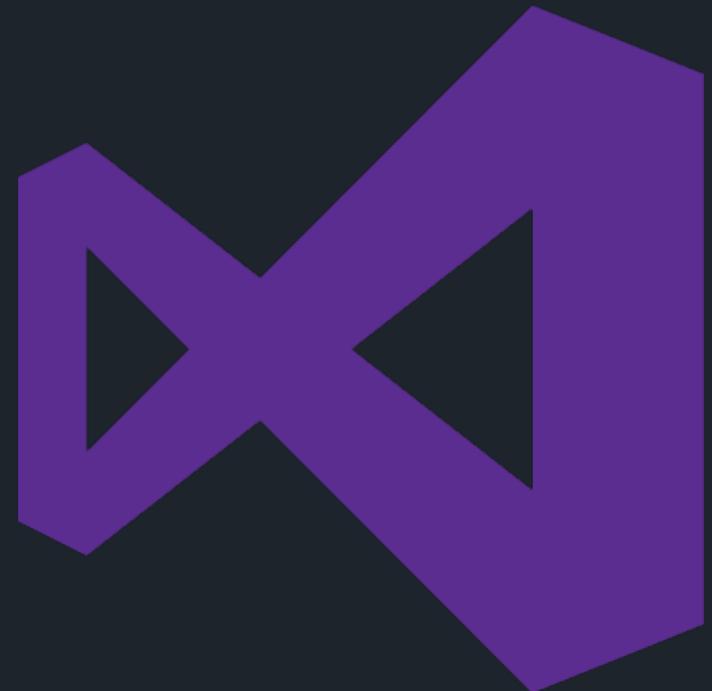
```
1 using Microsoft.AspNetCore.Hosting;
2 using Microsoft.Extensions.Configuration;
3 using Microsoft.Extensions.Hosting;
4 using Microsoft.Extensions.Logging;
5 using System;
6 using System.Collections.Generic;
7 using System.Linq;
8 using System.Threading.Tasks;
9
10 namespace LookingOverTheHedge.Api
11 {
12     public class Program
13     {
14         public static void Main(string[] args)
15         {
16             CreateHostBuilder(args).Build().Run();
17         }
18
19         public static IHostBuilder CreateHostBuilder(string[] args) =>
20             Host.CreateDefaultBuilder(args)
21                 .ConfigureWebHostDefaults(webBuilder =>
22                     {
23                         webBuilder.UseStartup<Startup>();
24                     });
25     }
26 }
27
```

The `Solution Explorer` pane on the right shows the project structure for `LookingOverTheHedge.Api`, which includes files like `HealthController.cs`, `TalksController.cs`, `Data`, `Migrations`, `appsettings.json`, `Program.cs`, `Startup.cs`, and `WeatherForecast.cs`.

Transform Sample API



AXXES



Visual
Studio

Axes



AWS TOOLKIT

- Plug in
- Get started faster
- Boost productivity



Visual
Studio



Rider



Visual Studio Code

AXXES

AWS Toolkit

demo

The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

- File Menu:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q).
- Solution Explorer:** Shows the project structure for "LookingOverTheHedge.Demo1.Before".
- Code Editor:** Displays the `Startup.cs` file content.
- Status Bar:** Labeled "Ready".

```
1 using LookingOverTheHedge.Api.Data;
2 using Microsoft.AspNetCore.Builder;
3 using Microsoft.AspNetCore.Hosting;
4 using Microsoft.AspNetCore.HttpsPolicy;
5 using Microsoft.AspNetCore.Mvc;
6 using Microsoft.Data.SqlClient;
7 using Microsoft.EntityFrameworkCore;
8 using Microsoft.Extensions.Configuration;
9 using Microsoft.Extensions.DependencyInjection;
10 using Microsoft.Extensions.Hosting;
11 using Microsoft.Extensions.Logging;
12 using System;
13 using System.Collections.Generic;
14 using System.Linq;
15 using System.Threading.Tasks;
16
17 namespace LookingOverTheHedge.Api
18 {
19     public class Startup
20     {
21         public Startup(IConfiguration configuration)
22         {
23             Configuration = configuration;
24         }
25
26         public IConfiguration Configuration { get; }
27
28         // This method gets called by the runtime. Use this method to add services to the container.
29         public void ConfigureServices(IServiceCollection services)
30         {
31             services.AddControllers();
32             var connectionStringBuilder = new SqlConnectionStringBuilder();
33             connectionStringBuilder.ConnectionString = Configuration.GetConnectionString("ConferenceConnection");
34             services.AddDbContext<ConferenceContext>(options => options.UseSqlServer(connectionStringBuilder.ToString()));
35         }
36
37         // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
38         public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
39         {
40             if (env.IsDevelopment())
41             {
42                 app.UseDeveloperExceptionPage();
43             }
44
45             app.UseHttpsRedirection();
46
47             app.UseRouting();
48
49             app.UseAuthorization();
50         }
51     }
52 }
```

The code in `Startup.cs` is responsible for setting up an ASP.NET Core application. It includes configurations for controllers, a database context (ConferenceContext), and various middleware components like routing, authorization, and developer exception pages.

Step1: Move data to the cloud

Database storage on AWS

Open minded

- Relational (RDS)
- Document
- Graph
- Time Series

Axes

Database services

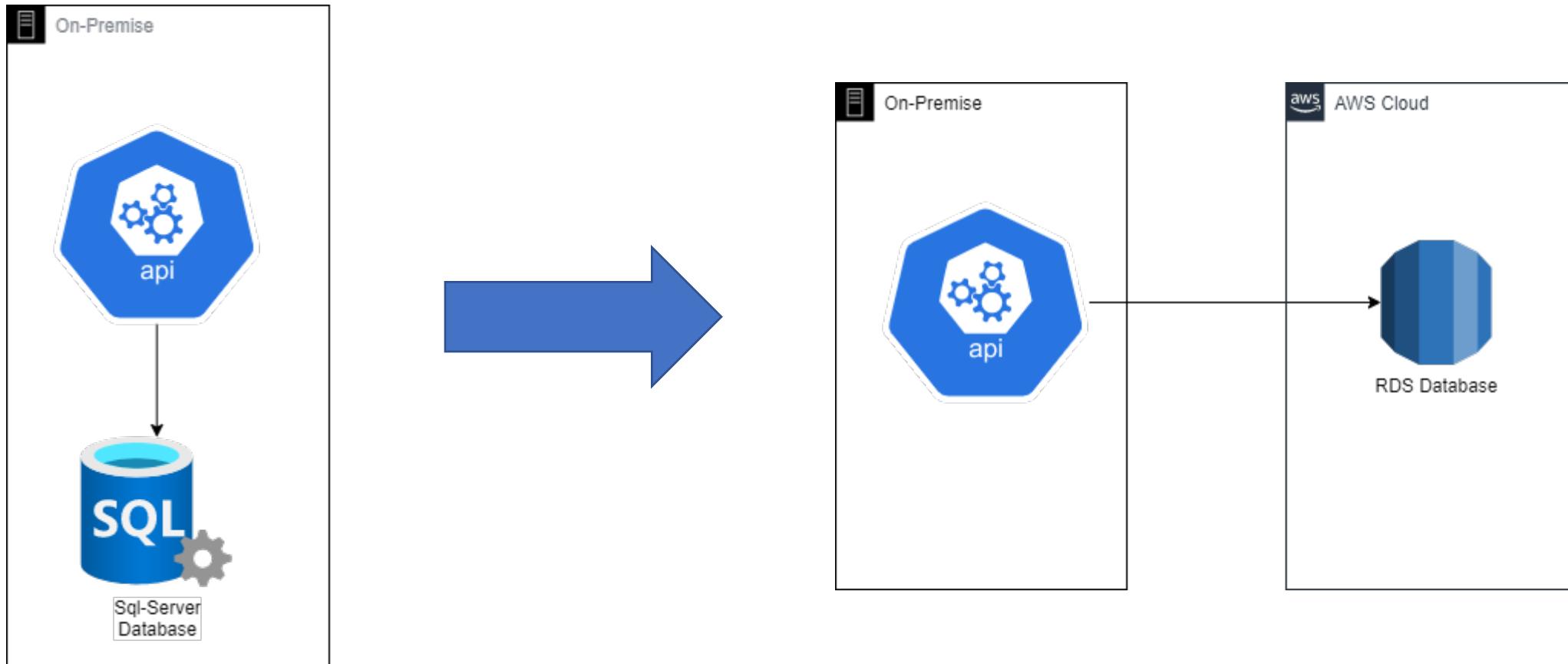
Database type	Use cases	AWS service
Relational	Traditional applications, ERP, CRM, e-commerce	 Amazon Aurora  Amazon RDS  Amazon Redshift
Key-value	High-traffic web apps, e-commerce systems, gaming applications	 Amazon DynamoDB
In-memory	Caching, session management, gaming leaderboards, geospatial applications	 Amazon ElastiCache for Memcached  Amazon ElastiCache for Redis  Amazon MemoryDB for Redis
Document	Content management, catalogs, user profiles	 Amazon DocumentDB (with MongoDB compatibility)
Wide column	High scale industrial apps for equipment maintenance, fleet management, and route optimization	 * Amazon Keyspaces (for Apache Cassandra)
Graph	Fraud detection, social networking, recommendation engines	 Amazon Neptune
Time series	IoT applications, DevOps, industrial telemetry	 Amazon Timestream
Ledger	Systems of record, supply chain, registrations, banking transactions	 Amazon QLDB

RDS database engines



- Free
 - PostgreSQL
 - MySQL
 - MariaDB
- Proprietary
 - Oracle
 - SQL Server
- Aurora
 - Serverless !

Step1: Move data to the cloud



AXXES_

Setup & usage RDS

demo

The screenshot shows the Microsoft Visual Studio IDE interface. The main area displays the `Program.cs` file for the project `LookingOverTheHedge.Api`. The code implements a `Program` class with a `Main` method that uses `IHostBuilder` to configure and run a web host.

```
1  using Microsoft.AspNetCore.Hosting;
2  using Microsoft.Extensions.Configuration;
3  using Microsoft.Extensions.Hosting;
4  using Microsoft.Extensions.Logging;
5  using System;
6  using System.Collections.Generic;
7  using System.Linq;
8  using System.Threading.Tasks;
9
10 namespace LookingOverTheHedge.Api
11 {
12     public class Program
13     {
14         public static void Main(string[] args)
15         {
16             CreateHostBuilder(args).Build().Run();
17         }
18
19         public static IHostBuilder CreateHostBuilder(string[] args) =>
20             Host.CreateDefaultBuilder(args)
21                 .ConfigureWebHostDefaults(webBuilder =>
22                     {
23                         webBuilder.UseStartup<Startup>();
24                     });
25     }
26 }
27
```

The Solution Explorer on the right shows the project structure with files like `appsettings.json`, `Program.cs`, `Startup.cs`, and `WeatherForecast.cs`.

Bottom status bar: 100% No issues found, Ln: 13 Ch: 6 SPC CRLF, Solution Explorer, Git Changes.

Bottom navigation: Error List, Developer PowerShell, Output.



AMAZON RDS

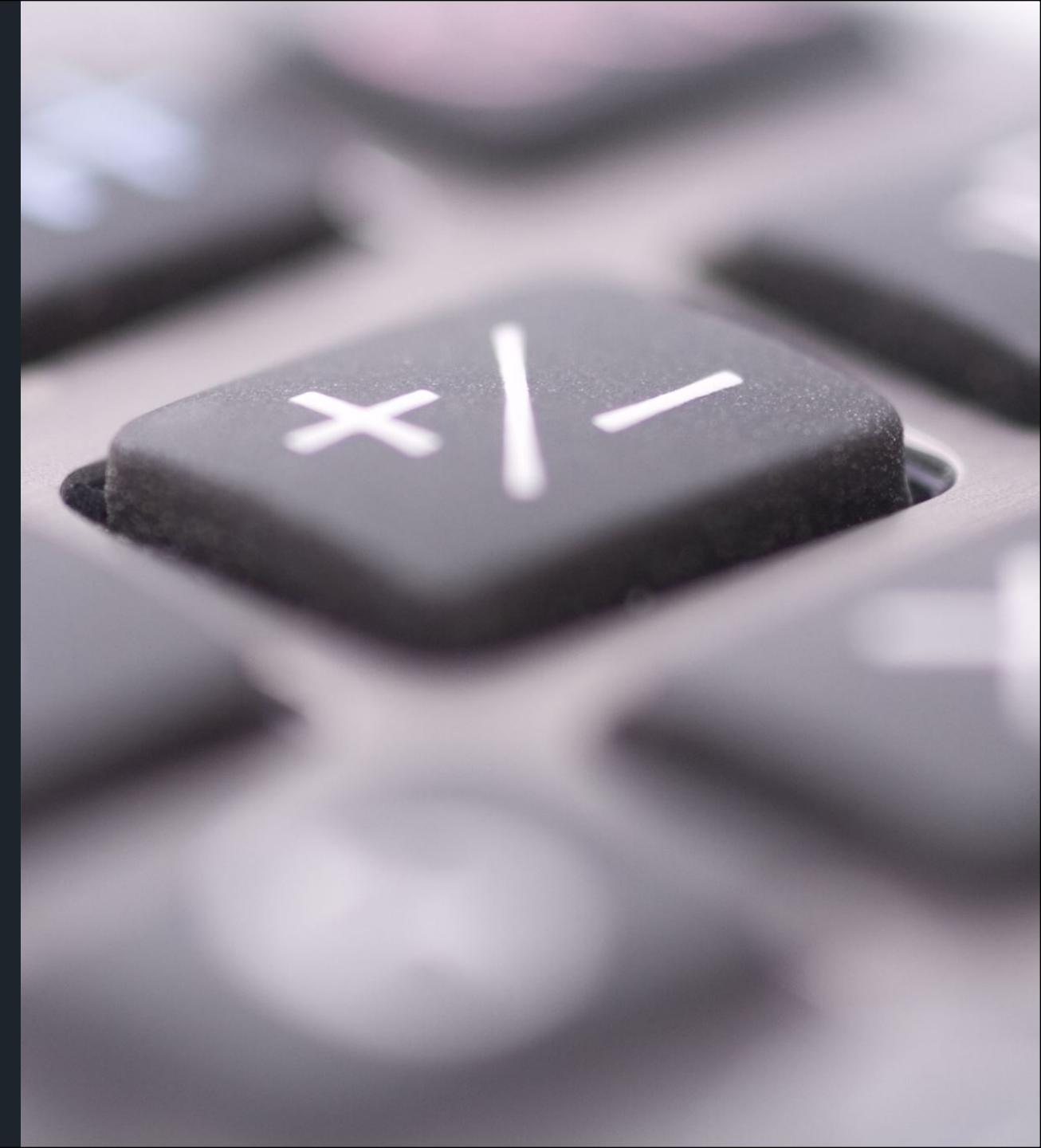


```
".ConnectionStrings": {  
    "RDSConnection": "Server=overthehedge.ckrgn3q2choj.eu-west-  
1.rds.amazonaws.com,1433;Database=ConferenceDatabase"  
}
```

Step2: Move Api to the cloud

Before moving forward:
some basics

- S3
 - Simple Storage Service
- EC2
 - Elastic Cloud Compute



Axes

'Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance.'

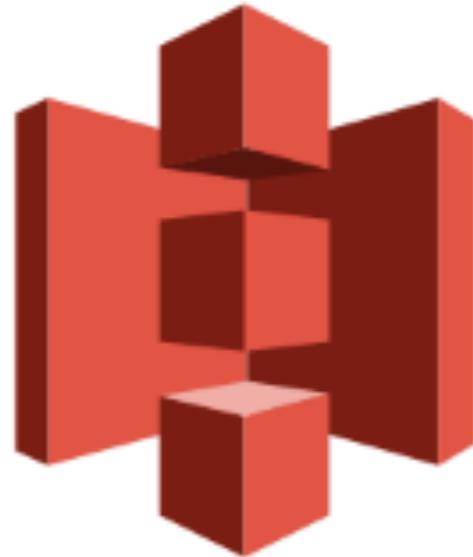
—

Simple Storage Service – S3

S3 provides 5GB of free standard storage

Simple Storage Service S3

- Data storage
- SPA Website / Cloudfront (CDN)
- Backup & Restore
- IoT
- Data Lakes / Big Data
- Pre-signed Urls
- ...



Amazon S3

‘Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. Amazon EC2’s simple web service interface allows you to obtain and configure capacity with minimal friction’

—

Elastic Compute Cloud – EC2

Free tier provides 750 hours/month of EC2 for 12 months

Elastic Cloud Compute EC2

- Virtual machine in the cloud
- Windows / Linux
- Instance type for each purpose
memory intensive/ CPU intensive
- Organized in Virtual Private
Networks



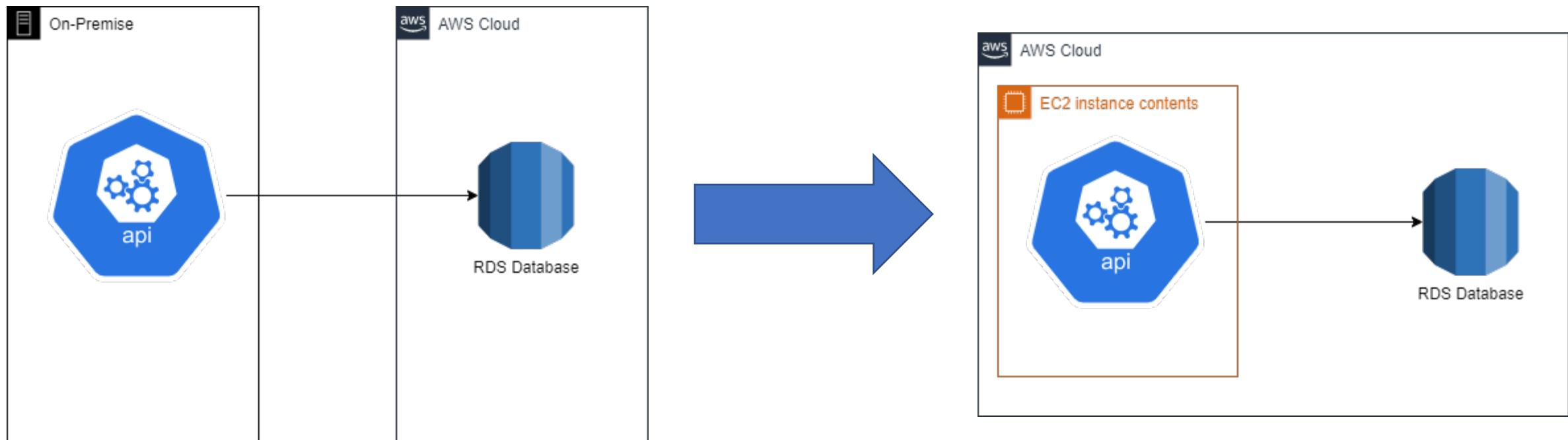
Amazon EC2



Elastic Beanstalk

- Simple way to deploy & scale your web-app
- Combination
 - EC2
 - S3
- No added costs

Step2: Move Api to the cloud



AXXES_

Elastic Beanstalk

demo

The screenshot shows the Microsoft Visual Studio IDE interface. The main area displays the code for `Startup.cs` in the `LookingOverTheHedge.Api` project. The code is annotated with tool tips and references, indicating the source of each line of code. The Solution Explorer on the right shows the project structure with files like `Program.cs`, `Startup.cs`, and `WeatherForecast.cs`. The bottom status bar shows the current line (Ln: 18), character (Ch: 2), and other settings.

```
Startup.cs  X
LookingOverTheHedge.Api  LookingOverTheHedge.Api.Startup  Startup(IConfiguration configuration)

1 using LookingOverTheHedge.Api.Data;
2 using Microsoft.AspNetCore.Builder;
3 using Microsoft.AspNetCore.Hosting;
4 using Microsoft.AspNetCore.HttpsPolicy;
5 using Microsoft.AspNetCore.Mvc;
6 using Microsoft.Data.SqlClient;
7 using Microsoft.EntityFrameworkCore;
8 using Microsoft.Extensions.Configuration;
9 using Microsoft.Extensions.DependencyInjection;
10 using Microsoft.Extensions.Hosting;
11 using Microsoft.Extensions.Logging;
12 using System;
13 using System.Collections.Generic;
14 using System.Linq;
15 using System.Threading.Tasks;
16
17 namespace LookingOverTheHedge.Api
18 {
19     2 references
20     public class Startup
21     {
22         0 references
23         public Startup(IConfiguration configuration)
24         {
25             Configuration = configuration;
26         }
27
28         4 references
29         public IConfiguration Configuration { get; }
30
31         // This method gets called by the runtime. Use this method to add services to the container.
32         0 references
33         public void ConfigureServices(IServiceCollection services)
34         {
35             services.AddControllers();
36             var connectionStringBuilder = new SqlConnectionStringBuilder();
37             connectionStringBuilder.ConnectionString = Configuration.GetConnectionString("RdsConnection");
38             connectionStringBuilder.UserID = Configuration.GetValue<string>("UserId");
39             connectionStringBuilder.Password = Configuration.GetValue<string>("Password");
40             services.AddDbContext<ConferenceContext>(options => options.UseSqlServer(connectionStringBuilder.ToString()));
41
42         // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
43         0 references
44         public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
45         {
46             if (env.IsDevelopment())
47             {
48                 app.UseDeveloperExceptionPage();
49
49             app.UseHttpsRedirection();
49             app.UseRouting();
```

Solution Explorer

Search Solution Explorer (Ctrl+S)

Solution 'LookingOverTheHedge.Demo2.Before' (1 of 1 project)

- LookingOverTheHedge.Api
 - Connected Services
 - Dependencies
 - Properties
 - Controllers
 - Data
 - Migrations
 - appsettings.json
 - Program.cs
 - Startup.cs
 - WeatherForecast.cs

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help

Search (Ctrl+Q) LookingOverTheHedge.Demo2.Before

Ready Add to Source Control

System Manager

- Removing the credentials from code!
- AWS Service System Manager – Parameter Store
- Easy way to store configuration and secure items
- Easy integration: Nuget package: Amazon.Extensions.Configuration.SystemsManager

System Manager

Demo



Services ▾

Search for services, features, marketplace products, and docs

[Alt+S]



Rob Van Pamel ▾

Ireland ▾

Support ▾

AWS Management Console

AWS services

Recently visited services

- Elastic Beanstalk
- RDS
- CloudWatch
- Lambda

- AWS Organizations
- API Gateway
- IAM
- CloudFormation

- S3
- Billing
- DynamoDB
- VPC

- Elastic Container Service
- Elastic Container Registry
- Systems Manager

All services

Build a solution

Get started with simple wizards and automated workflows.

Launch a virtual machine

With EC2
2-3 minutes



Build a web app

With Elastic Beanstalk
6 minutes



Build using virtual servers

With Lightsail
1-2 minutes



Register a domain

With Route 53
3 minutes



Connect an IoT device

With AWS IoT
5 minutes



Start migrating to AWS

With AWS MGN
1-2 minutes



Start a development project

With CodeStar
5 minutes



Deploy a serverless microservice

With Lambda, API Gateway
2 minutes



Stay connected to your AWS resources on-the-go



AWS Console Mobile App now supports four additional regions. Download the AWS Console Mobile App to your iOS or Android mobile device. [Learn more](#)

Explore AWS

AWS Training

Free digital courses to help you develop your skills.
[Learn more](#)

AWS Certification

Propel your career forward with AWS Certification.
[Learn more](#)

Free AWS Training

Advance your career with AWS Cloud Practitioner Essentials—a free, six-hour, foundational course. [Learn more](#)

AWS Certification Resources

Explore the resources available to help you prepare for your AWS Certification. [Learn more](#)

Have feedback?

AWS System Manager



```
public static IHostBuilder CreateHostBuilder(string[] args) =>
    Host.CreateDefaultBuilder(args)
        .ConfigureAppConfiguration((context, builder) =>
    {
        builder.AddSystemsManager("/OverTheHedge");
    })
        .ConfigureWebHostDefaults(webBuilder =>
    {
        webBuilder.UseStartup<Startup>();
    });
}
```

Step3: Containerize our API



Containers in AWS

Registry

- ECR

Orchestration

- ECS
- EKS
- Host your own ..

AWS ECR

Elastic Container Registry

- Managed registry for containers
- Storage
- Management



AXXES



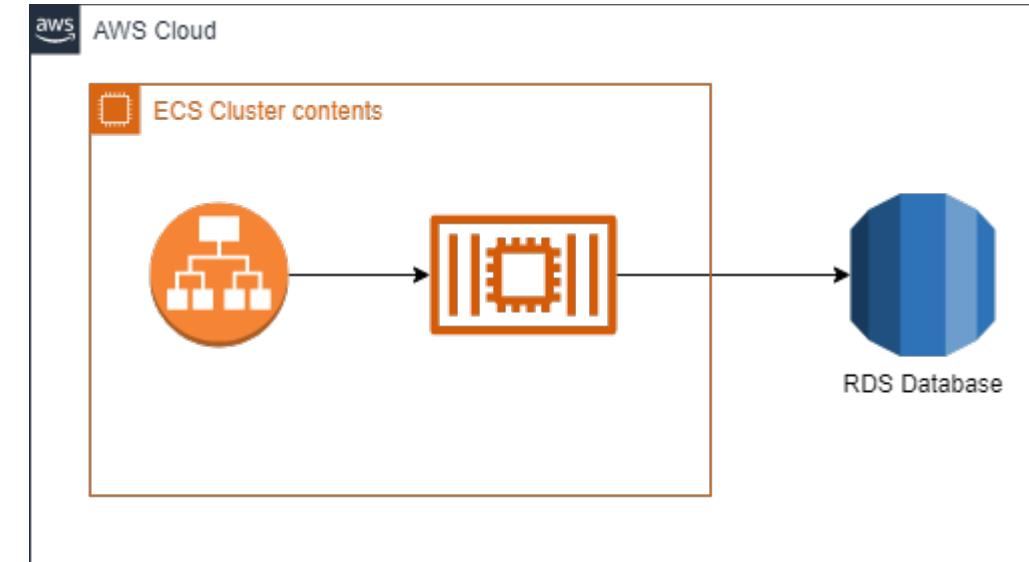
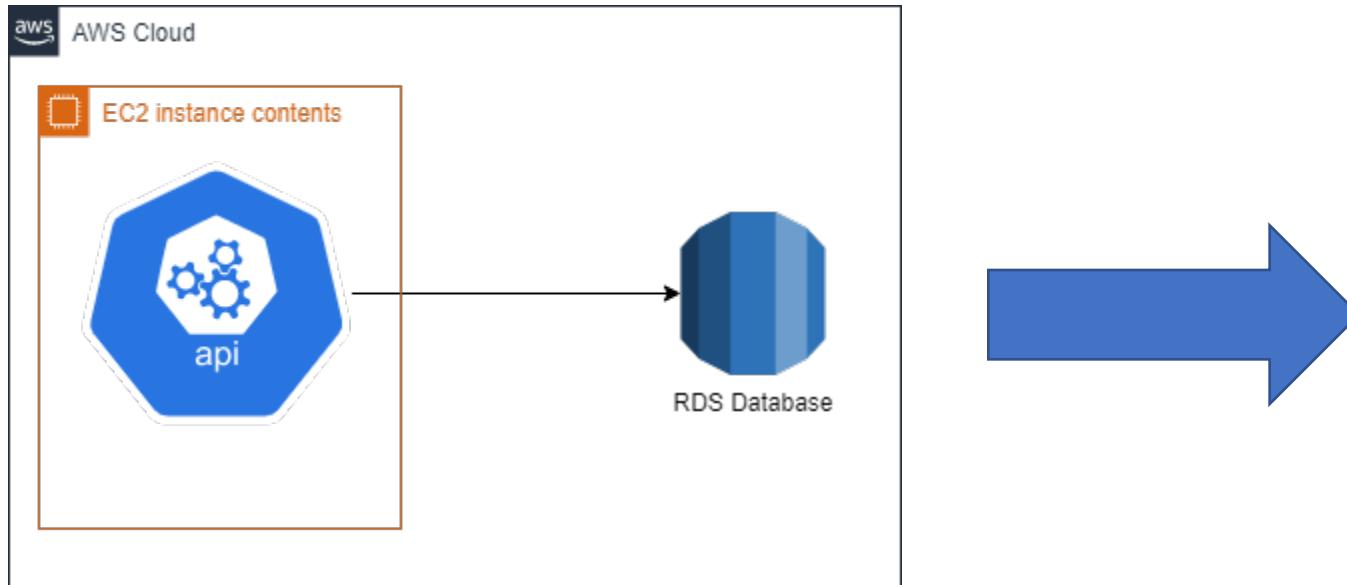
AWS ECS

- Elastic Container Service
- Secure & scalable container orchestration
- Tasks
 - A way to start a docker container. defined in a task definition
- Service
 - A automated way to start and scale tasks.
- Clusters
 - A bundle of services and tasks
- EC2 & Fargate

AWS EKS

- Elastic Kubernetes Service

Step3: Containerize our API



AXXES_

Containers on AWS

demo

The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

- Menu Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q)
- Toolbar:** Includes icons for Back, Forward, Home, Refresh, and others.
- Solution Explorer:** Shows the solution 'LookingOverTheHedge.Demo3.Before' with one project 'LookingOverTheHedge.Api'. The project structure includes Connected Services, Dependencies, Properties, Controllers, Data, Migrations, appsettings.json, aws-beanstalk-tools-defaults.json, Program.cs, Startup.cs, and WeatherForecast.cs.
- Code Editor:** The main editor window displays the 'Program.cs' file for the 'LookingOverTheHedge.Api' project. The code implements a host builder for an ASP.NET Core application, specifically adding a systems manager for the '/OverTheHedge' endpoint.
- Status Bar:** Shows '100 %' completion, 'No issues found', and file navigation information (Ln: 1 Ch: 1 SPC CRLF).
- Bottom Navigation:** Includes tabs for Error List, Developer PowerShell, Output, Solution Explorer, and Git Changes. A 'Ready' status indicator is also present.

```
1 using Microsoft.AspNetCore.Hosting;
2 using Microsoft.Extensions.Configuration;
3 using Microsoft.Extensions.Hosting;
4 using Microsoft.Extensions.Logging;
5 using System;
6 using System.Collections.Generic;
7 using System.Linq;
8 using System.Threading.Tasks;
9
10 namespace LookingOverTheHedge.Api
11 {
12     public class Program
13     {
14         public static void Main(string[] args)
15         {
16             CreateHostBuilder(args).Build().Run();
17         }
18
19         public static IHostBuilder CreateHostBuilder(string[] args) =>
20             Host.CreateDefaultBuilder(args)
21             .ConfigureAppConfiguration((context, builder) =>
22             {
23                 builder.AddSystemsManager("/OverTheHedge");
24             })
25             .ConfigureWebHostDefaults(webBuilder =>
26             {
27                 webBuilder.UseStartup<Startup>();
28             });
29     }
30 }
31
```

Step4: Move to serverless in the cloud

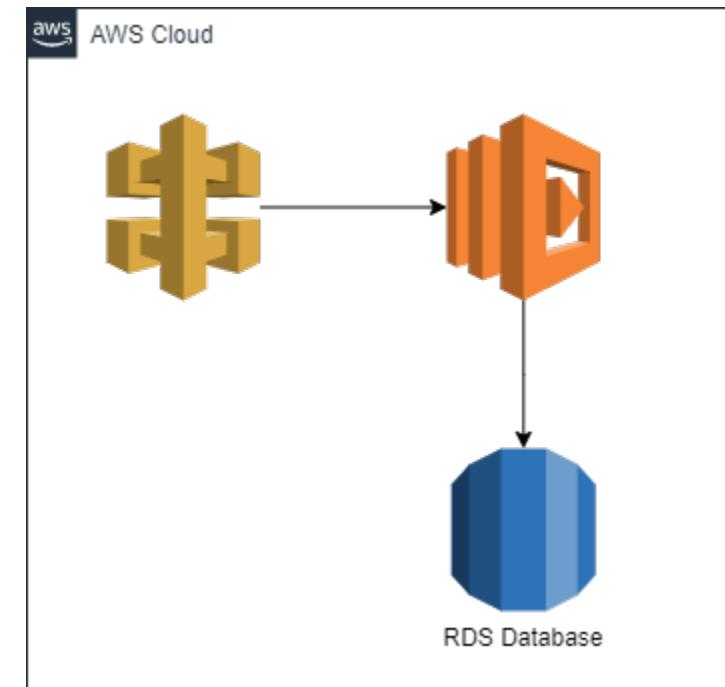
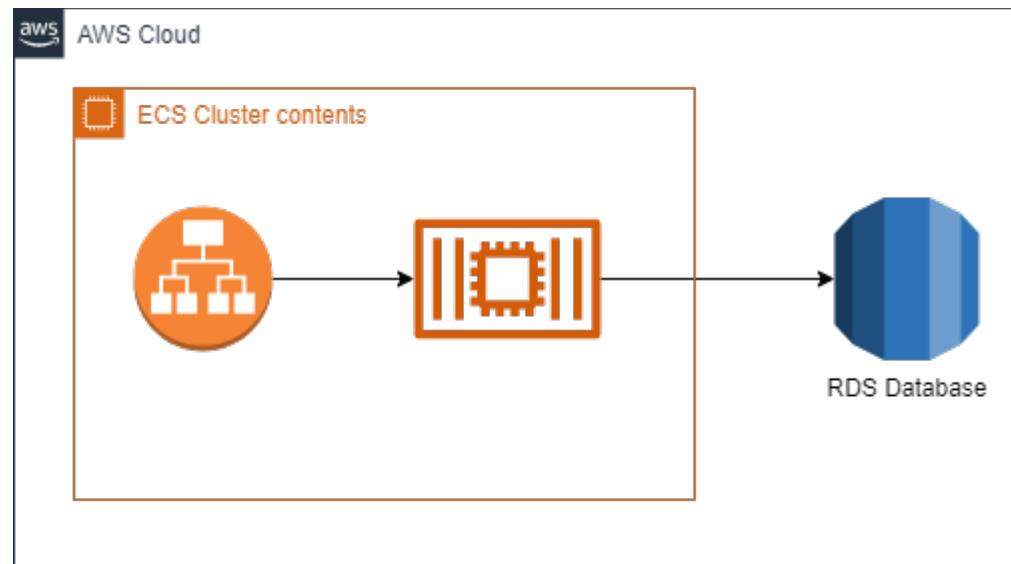
Run code without thinking about servers!

- No servers involved, just code!
- Code get triggered by event (eg api gateway)
- Cost optimized (per ms)
- "Unlimited" scaling
- Harder to test



Amazon
Lambda

Step4: transition to serverless



AXXES_

Lambda

—

Demo

The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

- Menu Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q).
- Toolbar:** Standard icons for file operations like Open, Save, Print, etc.
- Solution Explorer:** Shows the solution 'LookingOverTheHedge.Demo4.Before' with one project 'LookingOverTheHedge.Api'. The project structure includes Connected Services, Dependencies, Properties, Controllers, Data, Migrations, appsettings.json, Program.cs, Startup.cs, and WeatherForecast.cs.
- Code Editor:** The main window displays the 'Program.cs' file for the 'LookingOverTheHedge.Api' project. The code implements a host builder for an ASP.NET Core application, specifically targeting the 'OverTheHedge' system manager.
- Status Bar:** Shows the line number (Ln: 12), character position (Ch: 25), and file endings (SPC, CRLF).
- Bottom Navigation:** Error List, Developer PowerShell, Output, Solution Explorer, Git Changes.

```
1 using Microsoft.AspNetCore.Hosting;
2 using Microsoft.Extensions.Configuration;
3 using Microsoft.Extensions.Hosting;
4 using Microsoft.Extensions.Logging;
5 using System;
6 using System.Collections.Generic;
7 using System.Linq;
8 using System.Threading.Tasks;
9
10 namespace LookingOverTheHedge.Api
11 {
12     public class Program
13     {
14         public static void Main(string[] args)
15         {
16             CreateHostBuilder(args).Build().Run();
17         }
18
19         public static IHostBuilder CreateHostBuilder(string[] args) =>
20             Host.CreateDefaultBuilder(args)
21             .ConfigureAppConfiguration((context, builder) =>
22                 {
23                     builder.AddSystemsManager("/OverTheHedge");
24                 })
25             .ConfigureWebHostDefaults(webBuilder =>
26                 {
27                     webBuilder.UseStartup<Startup>();
28                 });
29     }
30 }
31
```

A status bar at the bottom indicates: This item does not support previewing.

Demo

```
public class LambdaEntryPoint : Amazon.Lambda.AspNetCoreServer.APIGatewayProxyFunction
{
    protected override void Init(IWebHostBuilder builder)
    {
        builder
            .ConfigureAppConfiguration((context, builder) =>
            {
                builder.AddSystemsManager("/OverTheHedge");
            })
            .UseStartup<Startup>();
    }
}
```

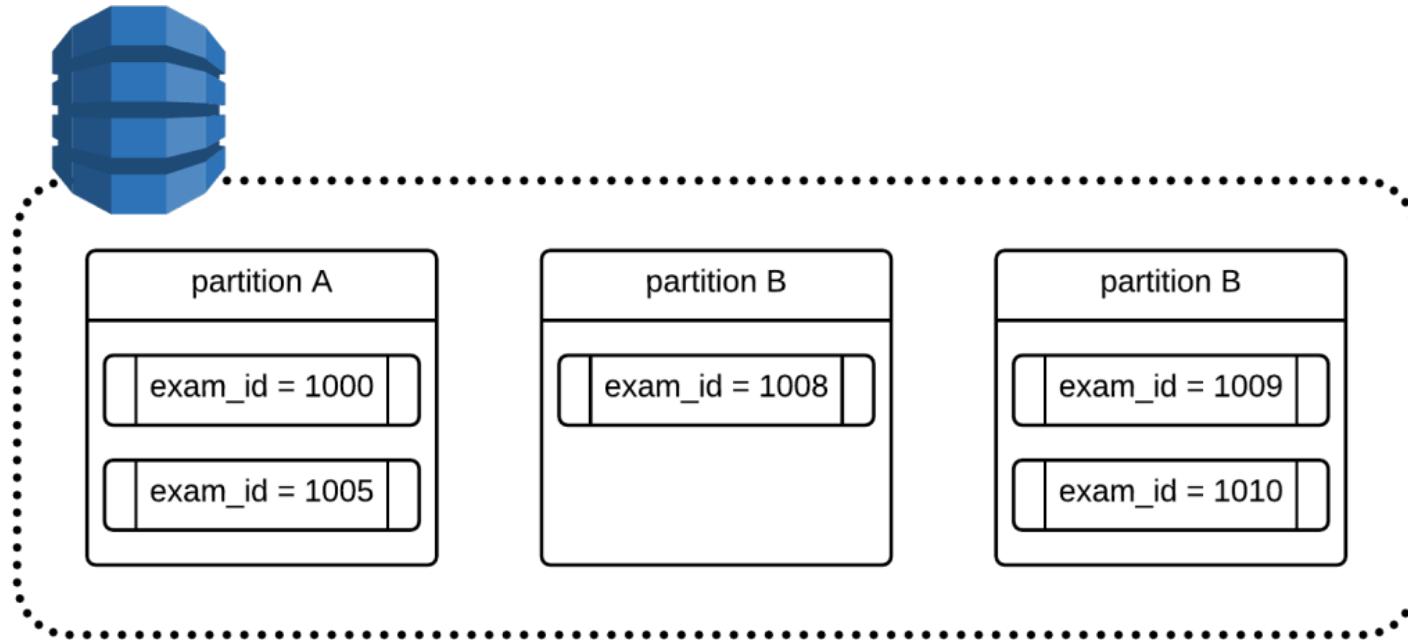
Step5: Use cloud native storage

DynamoDB

- NoSql database @ scale
- Single-digit millisecond
- Key-Value & document data models
- PartiQL compatible
- Transaction support
- Blends with
 - Micro-services
 - Mobile back-ends
 - Serverless Apps

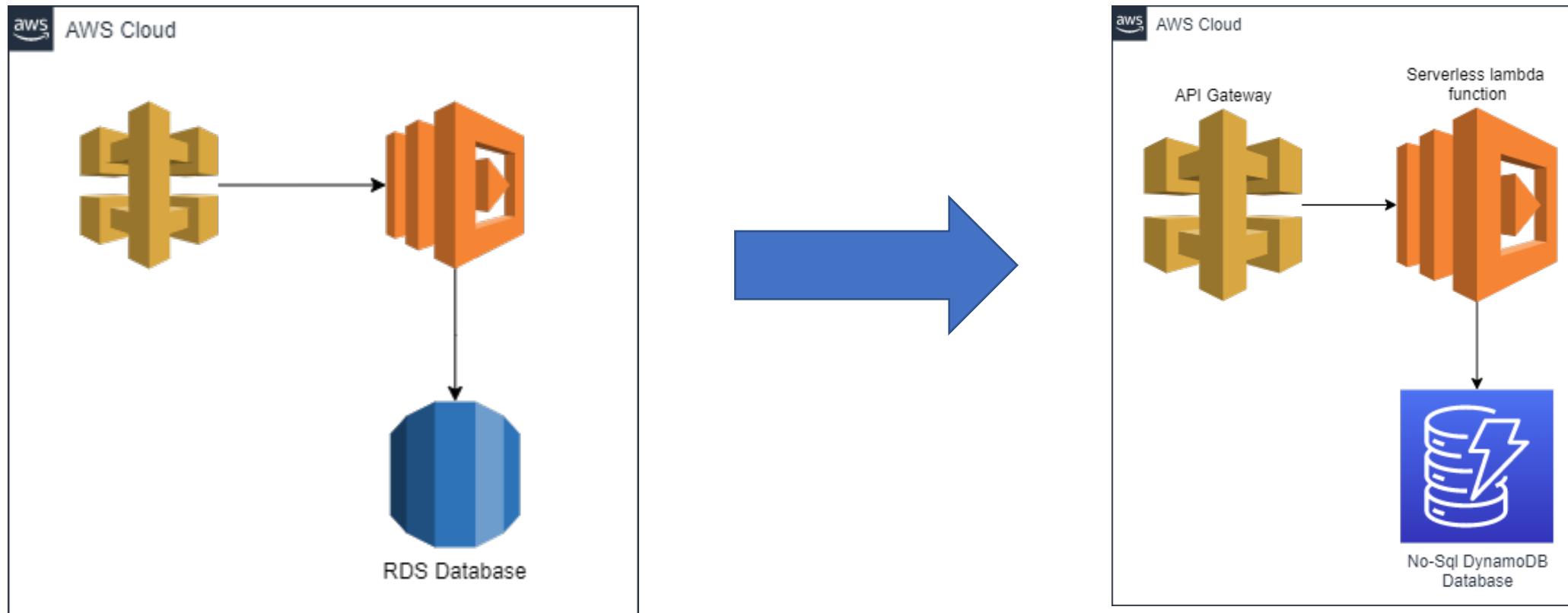
DynamoDB characteristics

- Tables
- Primary Key
 - Partition key (Hash key)
 - *Sort key (Range Key)*
- Secondary Indexes
 - Global
 - Local
- Query vs scans
- Streams
- Take time to learn it!



AXES

Step5: Use cloud native storage



AXXES_

DynamoDB

Demo

AWS Management Console

AWS services

Recently visited services

[Lambda](#)[IAM](#)[Systems Manager](#)[S3](#)[CloudWatch](#)[CloudFormation](#)[RDS](#)[Billing](#)[VPC](#)[Elastic Beanstalk](#)[AWS Organizations](#)[DynamoDB](#)[EC2](#)[Elastic Container Service](#)[API Gateway](#)

All services

Build a solution

Get started with simple wizards and automated workflows.

Launch a virtual machine

With EC2

2-3 minutes



Build a web app

With Elastic Beanstalk

6 minutes



Build using virtual servers

With Lightsail

1-2 minutes



Register a domain

With Route 53

3 minutes



Connect an IoT device

With AWS IoT

5 minutes



Start migrating to AWS

With AWS MGN

1-2 minutes



Start a development project

With CodeStar

5 minutes



Deploy a serverless microservice

With Lambda, API Gateway

2 minutes



Stay connected to your AWS resources on-the-go



AWS Console Mobile App now supports four additional regions. Download the AWS Console Mobile App to your iOS or Android mobile device. [Learn more](#)

Explore AWS

AWS Certification

Propel your career forward with AWS Certification.

[Learn more](#)

AWS Machine Learning Training

Choose from courses that cover the entire machine learning pipeline. [Learn more](#)

Free AWS Training

Advance your career with AWS Cloud Practitioner Essentials—a free, six-hour, foundational course. [Learn more](#)

Amazon Location Service

Easily and securely add maps, search for points of interest, geocoding, routes, tracking, and geofencing to your application. [Get started](#)

Demo



```
var dbClient = new Amazon.DynamoDBv2.AmazonDynamoDBClient();
DynamoDBContext context = new DynamoDBContext(dbClient);

// Create a query to fetch all the talks from the conference
DynamoDBOperationConfig operation = new DynamoDBOperationConfig();
operation.QueryFilter.Add(new ScanCondition("SK",
Amazon.DynamoDBv2.DocumentModel.ScanOperator.BeginsWith, "TALK#"));

var query = context.QueryAsync<Talk>("CONFERENCE#LiveOnStage", operation);
var talks = new List<Talk>();
do
{
    var resultSet = await query.GetNextSetAsync();
    talks.AddRange(resultSet);
} while (!query.IsDone);
```

AWS & .NET

- S3
- SQS / SNS
- Cognito (Identity Provider)
-
- AWS Re-invent 2021 (29/11 – 3/12)



AXXES

Resources

- Samples on <https://github.com/robvanpamel/dotnet-on-aws/>
- <https://aws.amazon.com/training/>
- <https://aws.amazon.com/developer/language/net/>
- ...

Axes

Thank you!



codit

involved



team + talent



Axes

