

Fontys University of Applied Sciences

Cloud Resources

GetawayGo

Anna Kadurin
18/01/2025

Table of Contents

Resource groups	1
App Services.....	1
Static web apps	2
SQL Servers	2
SQL Databases	3
Azure Service Bus	4
Azure Load Testing	4
Resource group with its resources	5
Azure DevOps Connections to the Azure Resources	5

Resource groups

I have decided on creating separate resources group for each of the microservices and the frontend. This way I can easily separate concerns.

Home >		
Resource groups		
Office 365 Fontys (stichtingfontys.onmicrosoft.com)		
+ Create Manage view Refresh Export to CSV Open query Assign tags		
Filter for any field... Subscription equals all Location equals all Add filter		
Showing 1 to 11 of 11 records.		
No grouping List view		
<input type="checkbox"/> Name ↑↓	Subscription ↑↓	Location ↑↓
<input type="checkbox"/> DefaultResourceGroup-WEU	AzureSubscriptionSchool	West Europe
<input type="checkbox"/> GetawayGo-rg-analyticsservice	AzureSubscriptionSchool	West Europe
<input type="checkbox"/> GetawayGo-rg-apigateway	AzureSubscriptionSchool	West Europe
<input type="checkbox"/> GetawayGo-rg-bookingservice	AzureSubscriptionSchool	West Europe
<input type="checkbox"/> GetawayGo-rg-chatservice	AzureSubscriptionSchool	West Europe
<input type="checkbox"/> GetawayGo-rg-loadtesting	AzureSubscriptionSchool	West Europe
<input type="checkbox"/> GetawayGo-rg-NotificationsService	AzureSubscriptionSchool	West Europe
<input type="checkbox"/> GetawayGo-rg-property-service	AzureSubscriptionSchool	West Europe
<input type="checkbox"/> GetawayGo-rg-reviewservice	AzureSubscriptionSchool	West Europe
<input type="checkbox"/> GetawayGo-rg-userservice	AzureSubscriptionSchool	West Europe
<input type="checkbox"/> GetawayGoFrontend	AzureSubscriptionSchool	West Europe
< Previous Page 1 of 1 Next >		
Give feedback		

Figure 1 – Resource groups

App Services

Each microservice is deployed in an App Service or Web App. These resources are easy to manage and are automatically scaled. Each service is automatically deployed via its pipeline in Azure DevOps.

Home >

App Services

Office 365 Fontys (stichtingfontys.onmicrosoft.com)

+ Create Manage Deleted Apps Manage view Refresh Export to CSV Open query Assign tags Start Restart Stop Delete

Filter for any field... Subscription equals all Resource group equals all Location equals all Add filter

Showing 1 to 8 of 8 records.

No grouping List view

<input type="checkbox"/> Name ↑↓	Status ↑↓	Location ↑↓	Pricing Tier ↑↓	App Service Plan ↑↓	Subscription ↑↓	App Type ↑↓
<input type="checkbox"/> AnalyticsServiceGetawayGo	Running	West Europe	Free	GetawayGoServicePlan	AzureSubscriptionSchool	Web App ...
<input type="checkbox"/> APIGatewayGetawayGo	Running	West Europe	Free	GetawayGoServicePlan	AzureSubscriptionSchool	Web App ...
<input type="checkbox"/> BookingServiceGetawayGo	Running	West Europe	Free	GetawayGoServicePlan	AzureSubscriptionSchool	Web App ...
<input type="checkbox"/> ChatServiceGetawayGo	Running	West Europe	Free	GetawayGoServicePlan	AzureSubscriptionSchool	Web App ...
<input type="checkbox"/> GetawayGoReviewService	Running	West Europe	Free	GetawayGoServicePlan	AzureSubscriptionSchool	Web App ...
<input type="checkbox"/> NotificationServiceGetawayGo	Running	West Europe	Free	GetawayGoServicePlan	AzureSubscriptionSchool	Web App ...
<input type="checkbox"/> PropertyServiceGetawayGo	Running	West Europe	Free	GetawayGoServicePlan	AzureSubscriptionSchool	Web App ...
<input type="checkbox"/> UserServiceGetawayGo	Running	West Europe	Free	GetawayGoServicePlan	AzureSubscriptionSchool	Web App ...

< Previous Page 1 of 1 Next >

[Give feedback](#)

Figure 2 – App Services

Static web apps

I have only one static web app – the frontend. Static web apps are extremely suitable for React apps and automatically handle loads of request, thus not needing a typical load balancer. The deployment of the frontend to the static web app is once again automatic via the Azure DevOps pipelines.

Home >

Static Web Apps

Office 365 Fontys (stichtingfontys.onmicrosoft.com)

+ Create Manage view Refresh Export to CSV Open query Assign tags

Filter for any field... Subscription equals all Resource group equals all Location equals all Add filter

Showing 1 to 1 of 1 records.

No grouping List view

<input type="checkbox"/> Name ↑↓	Type ↑↓	Resource group ↑↓	Location ↑↓	SKU ↑↓	Subscription ↑↓	Default hostname ↑↓
<input type="checkbox"/> getawaygo	Static Web App	GetawayGoFrontend	West Europe	Free	AzureSubscriptionSchool	nice-mushroom-09ebf5f03... ..

Figure 3 – Static Web Apps

SQL Servers

Due to costs, only 2 SQL servers are currently present in the Azure environment – one for the users and one for the properties.

Name	Status	Resource group	Location	Subscription
getawaygoproperties	Available	GetawayGo-rg-property	West Europe	AzureSubscriptionSchool
getawaygousers	Available	GetawayGo-rg-property	West Europe	AzureSubscriptionSchool

Figure 4 – SQL Servers

To login in the server, my account is required, as I have utilized Microsoft Entra Id to set myself as admin for security reasons. The related services to the servers are also added as admins with managed identity and communicate with a connection string which does not contain username and password. All connection strings are in variable groups as secret. This way the servers and databases are secure.

Microsoft Entra admin

Microsoft Entra authentication allows you to centrally manage identity and access to your Azure SQL Database. [Learn more](#)

Admin name: 493249@student.fontys.nl (Admin Object/App ID: ec6f5087-560a-40bb-a50f-427dd71314e9)

Figure 5 – Microsoft Entra ID of SQL Server

SQL Databases

Each server has one database with multiple tables based on the requirements.

Name	Server	Replica type	Pricing tier	Location	Subscription
getawaygoproperties (getawaygoproperties/getawaygoproperties)	getawaygo...	--	General Purpose: G...	West Europe	AzureSubscriptionSchool
getawaygousers (getawaygousers/getawaygousers)	getawaygou...	--	Basic: 5 DTUs	West Europe	AzureSubscriptionSchool

Figure 6 – SQL Databases

Azure Service Bus

I have created an Azure Service Bus to enable asynchronous communication between microservices. The bus has queues where the services can publish and subscribe to messages.

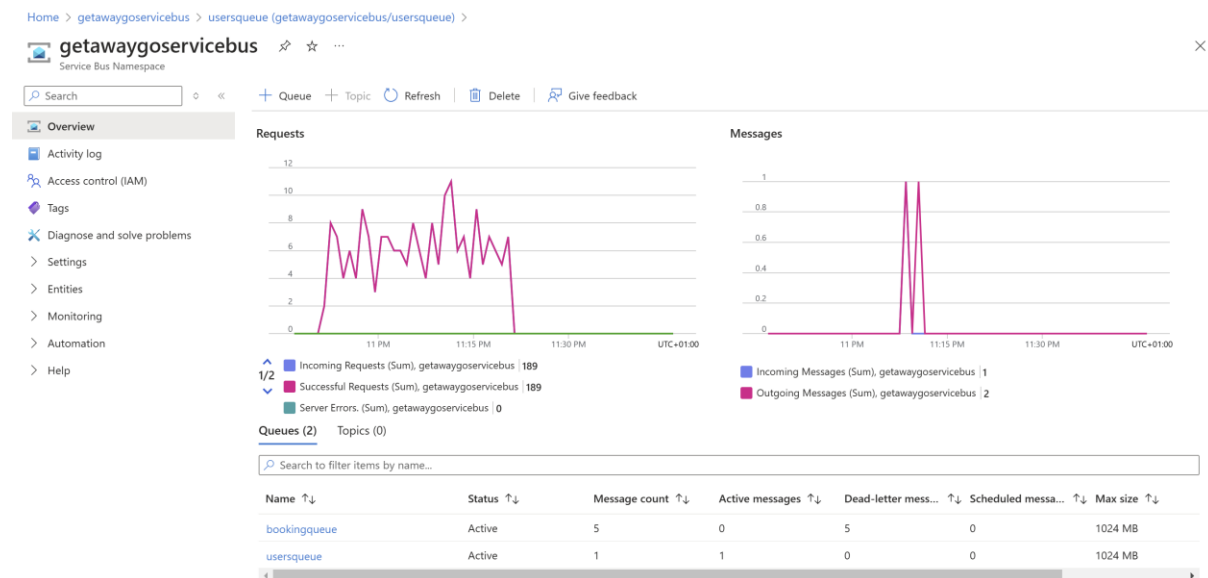


Figure 7 – Azure Service Bus

Azure Load Testing

One more resource that I am using from Azure is the Load Testing. This specific resource uses Apache Jmeter to perform tests based on a script. I have created 2 tests at the moment with multiple metrics.

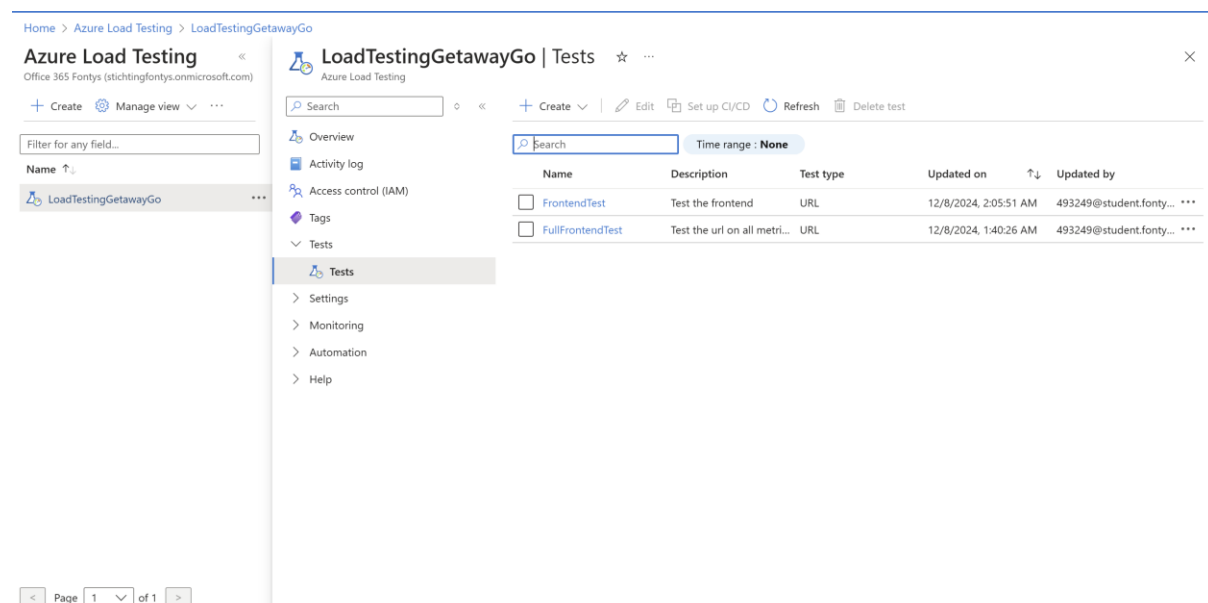


Figure 8 – Azure Load Testing

Resource group with its resources

An example resource group contains the App Service with Application Insights, as well as the SQL server and database. The app service plan that you see below is used by all resources for GetawayGo.

The screenshot shows the Azure portal interface for the resource group 'GetawayGo-rg-userservice'. The 'Resources' tab is selected, displaying a list of resources. The table below represents the data shown in the screenshot.

Name	Type	Location
Application Insights Smart Detection	Action group	Global
GetawayGoServicePlan	App Service plan	West Europe
getawaygousers	SQL server	West Europe
getawaygousers (getawaygousers/getawaygousers)	SQL database	West Europe
UserServiceGetawayGo	App Service	West Europe
UserServiceGetawayGo	Application Insights	West Europe

Figure 9 – Resource group example – users

Azure DevOps Connections to the Azure Resources

To be able to deploy all the microservices and frontend to their spots in Azure, a connection needs to be made via Azure DevOps. All these connections that you see below are connected with the relative App Service. The connections are Azure Resource Manager and utilize the automatic Service Principle. I use them via the pipelines.

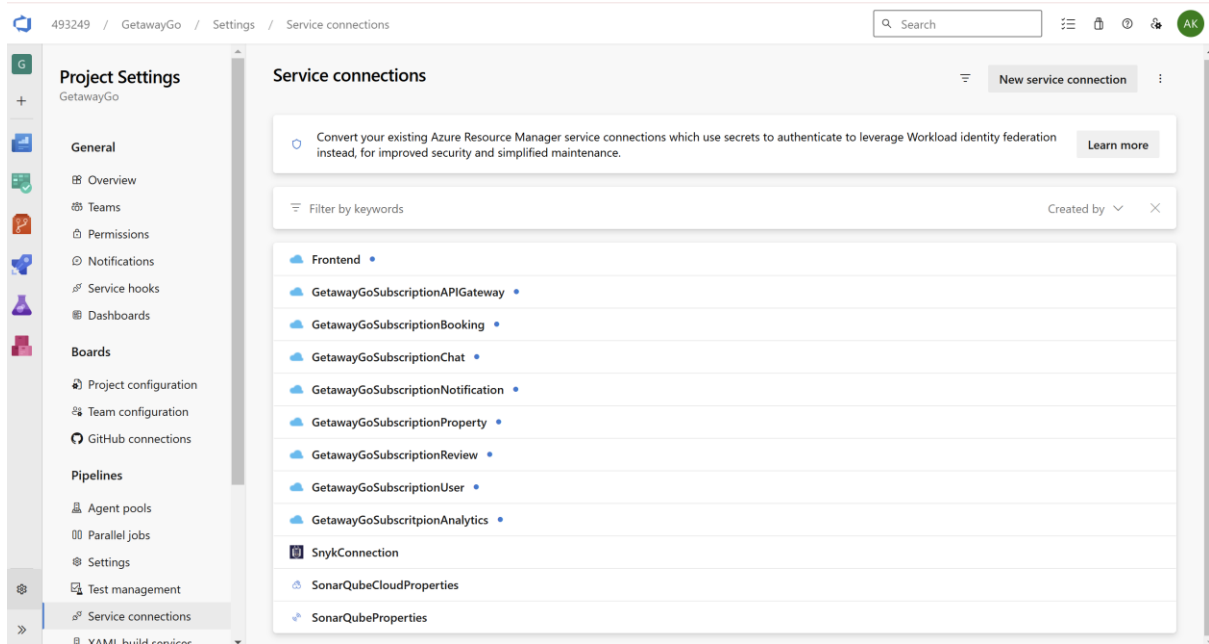


Figure 10 – Azure DevOps Service Connections