

2. laboratorijska vježba - zadatak

Cilj ove vježbe je napisati jednostavan program kako biste se upoznali s bibliotekama za rad s Azure Data Lake Gen2, te napraviti funkcije za čitanje i pisanje podataka iz/u *Data Lake*. Također, upoznat ćete se s *Azure Service Bus* koji je pub/sub agent za proslijeđivanje poruka. Sav programski kod i funkcije kreirane u ovoj vježbi, za komunikaciju s navedenim servisima, možete koristiti u idućim laboratorijskim vježbama kada ćete graditi podatkovni cjevovod (*engl. data pipeline*).

Za potrebe ove vježbe koristite verziju **Go 1.18** zbog rada s vanjskim bibliotekama. Svaki zadatak ima napomenu koja vanjska biblioteka se očekuje u rješenju. Kreirat ćete Azure Key Vault i Service Bus, te ćete učitati podatke na svoj Data Lake pomoć programa napisanog u Go-u.

Napomena: Prilikom svakog kreiranja novog resursa trebate podesiti role za svoju grupu. Kada kreirate resurs, u lijevom izborniku pronađite *Access Control (IAM)*. Sada kliknite *+Add Add role assignment*, te odaberite *Owner*. Kliknite *Next*, te potom u *members* odaberite svoju grupu. Kliknite *Review+assign Create*.

(Ukoliko ovaj korak niste napravili na prošloj vježbi prilikom kreiranja resursne grupe i aks-a napravite to sad)

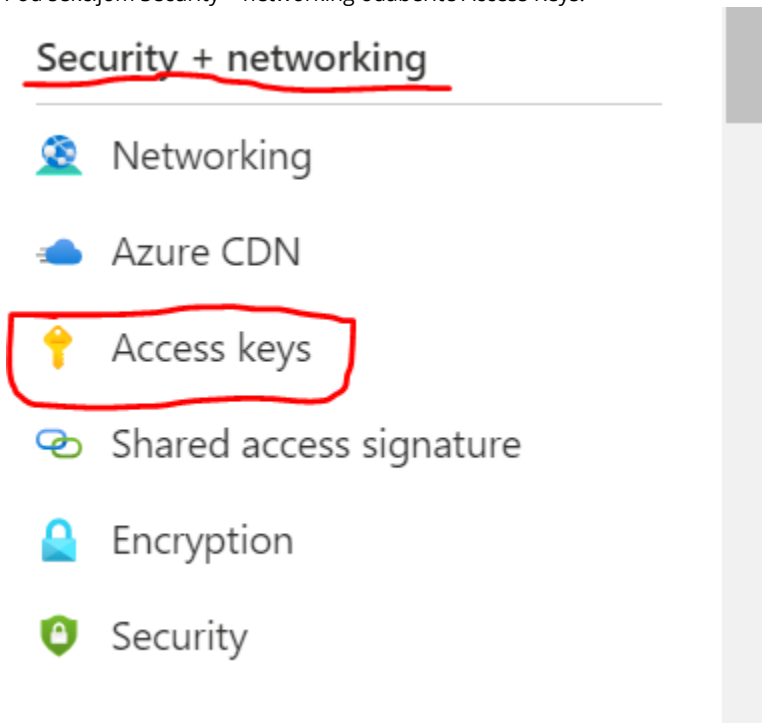
Čitanje/pisanje podataka iz spremišta podataka

U prvome dijelu laboratorijske vježbe konfigurirat ćete Azure Key Vault, kako biste se upoznali sa sigurnom pohranom lozinki i ostalih osjetljivih informacija (*engl. Secrets*), te ćete učitati datoteku na vaš *Data Lake* koji ste konfigurirali u prethodnoj vježbi.




Azure Key Vault konfiguracija

Kako biste pristupili svom spremištu podataka potrebno je napraviti iduće korake:

1. Pronađite svoj novokreirani Storage Account. Kliknite na njega.
2. Pod sekcijom Security + networking odaberite Access Keys.




3. Odaberite Show keys te kopirajte vrijednost prvog prikazanog ključa.

 Show keys  Set rotation reminder  Refresh

Access keys authenticate your applications' requests to this storage account. Keep your keys in a Key Vault, and replace them often with new keys. The two keys allow you to replace one while still

Remember to update the keys with any Azure resources and apps that use this storage account.

Storage account name

key1  Rotate key

Last rotated: 2/23/2022 (49 days ago)

Key

Connection string

4. Pronađite i kreirajte Azure Key Vault. Unesite potrebne podatke. Pritom se držite konvencije imenovanja: <naziv_tima>-<tip_resursa>-<naziv_aplikacije> (npr. Tim1-azv-ferlab).
5. Nakon unosa podataka i imenovanja resursa odaberite opciju "Create".
6. Nakon uspješnog kreiranja Key Vault-a, potrebno je dodati Secret.
 - a. Odaberite kreirani Key Vault
 - b. U izborniku odaberite Secrets te zatim Generate/Import. Upišite željeno ime Secreta i vrijednost ključa koju ste kopirali za svoj Storage Account.
 - c. Odaberite opciju "Create".

Key Vault Secrets i Credentials

1. Kako biste se mogli spojiti na Data Lake, potrebno je dohvatiti Secret za spajanje koji ste u prethodnom koraku spremili na Azure Key Vault.

Napišite funkciju `vaultInfo()` koja vraća vrijednost zadanog Secreta koji se nalazi u zadanom Key Vaultu.

 - Koristite sljedeće biblioteke: <https://pkg.go.dev/github.com/Azure/azure-sdk-for-go/sdk/azidentity> te <https://pkg.go.dev/github.com/Azure/azure-sdk-for-go/sdk/keyvault/azsecrets>.
2. U glavnom dijelu programa, koristeći biblioteku "<https://github.com/Azure/azure-sdk-for-go/tree/main/sdk/storage/azblob>" kreirajte credentials kako biste se autorizirali za rad s Azure Data Lake Gen 2. Iskoristite `vaultInfo` funkciju te dohvaćenu vrijednost Secret-a.

Napomena: Prije pokretanja programa, prijavite se u Azure portal putem Azure CLI-a za potrebe autentifikacije.

Napomena: Potrebnu konfiguraciju programa učitajte preko varijabli okoline.

Pisanje podataka

1. Preuzmite skup podataka sa sljedeće poveznice: <https://www.kaggle.com/datasets/jr2ngb/superstore-data>
2. Preuzetu datoteku koja sadrži podatke o narudžbama potrebno je prvo staviti na Azure Data Lake Storage Gen 2 koji ste konfigurirali u prethodnoj vježbi.

Pronađite svoj novokreirani Storage Account te odaberite opciju "+Container" kako biste kreirali kontejner, koji će služiti za grupiranje vaših podataka:

- a. Name Ime Storage kontejnera. Budući da će ovaj kontejner služiti kao zona gdje dolaze podaci iz nekog vanjskog sustava u Data Lake, prijedlog je da se nazove "landing".
- b. Public Access Level Private (no anonymous access)

c. Create

- Unutar glavnog dijela programa potrebno je kreirati objekte koji vam služe za interakciju s vašim Data Lake-om. Jedan od tih objekata je objekt tipa *service client*. Kreirajte ga koristeći "With Shared Key" opciju te koristite kreirani credential iz prethodnog dijela zadatka.
 - Koristite <https://github.com/Azure/azure-sdk-for-go/tree/main/sdk/storage/azblob> biblioteku.
- Nakon što ste kreirali Service Client objekt, potrebno je kreirati i *Container Client* koji služi za interakciju sa samim kontejnerom. Također je potrebno kreirati i objekt tipa *Blob Client* kojeg koristite za prijenos blob-ova podataka u svoj Data Lake. Unutar Storage kontejnera napravite posebni direktorij u koji ćete prenijeti podatke (npr. *retailData*).

Napomena: pripazite da prilikom kreiranja objekta predate ispravno ime svog kontejnera. Ako ste pratili upute od prije onda bi to bilo "landing".

- Kreirajte, te u glavnom dijelu programa pozovite funkciju: `uploadFile(filename string, blobClient *azblob.BlockBlobClient, ctx context.Context)`. Funkcija prima ime lokalne datoteke koju želite zapisati na Data Lake, prethodno kreirani blob klijent te GoLang kontekst.

Čitanje podataka

U ovom dijelu laboratorijske vježbe pročitati ćete sadržaj datoteke koju ste prethodno pomoću programa učitali na *Data Lake*. Za dohvaćanje vrijednosti Secret-a te kreiranje svih objekata za spajanje na *Data Lake* možete iskoristiti kod iz prethodnog dijela vježbe.

- Napišite funkciju `downloadData(blobClient *azblob.BlockBlobClient, ctx context.Context)` za dohvaćanje podataka iz *Data Lake-a*, koja vraća polje byte-ova (polje byte-ova potrebno je kako biste kasnije mogli kreirati poruke koje ćete slati na *Azure Service Bus*). Kao i u prethodnom dijelu zadataka, koristite <https://github.com/Azure/azure-sdk-for-go/tree/main/sdk/storage/azblob> biblioteku.
- Napišite funkciju koja će podatke iz formata za prijenos (`[]byte`) pretvoriti u odgovarajući oblik kako biste kasnije mogli kreirati poruke koje će se slati na *Azure Service Bus*.
 - Koristite sljedeći paket: "<http://github.com/gocarina/gocsv>"
 - Kreirajte strukturu `Order` čiji atributi odgovaraju stupcima .csv datoteke.
 - Napišite funkciju koja će preuzeto polje byte-ova, koje predstavlja sadržaj vaše datoteke, pretvoriti u polje čiji su elementi tipa `Order`

Kreiranje Service Bus topica

Kreirajte Service Bus Namespace:

Unutar svoje grupe resursa odaberite opciju Create i zatim kliknite *Service Bus*. Prije nego što kliknete *Review + Create*, ispunite tražene informacije na sljedeći način:

- Subscription vaša aktivna pretplata na Azure-u
- Resource group vaša grupa resursa
- Namespace name <ime_tima>-<tip_resursa>-<naziv_aplikacije> (npr. Tim1-sb-fer)
- Location West Europe
- Pricing tier Standard



Create namespace ...

Service Bus

Basics Networking Tags Review + create

Project Details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource group *

[Create new](#)

Instance Details

Enter required settings for this namespace.

Namespace name * .servicebus.windows.net

Location *

Pricing tier *

[Browse the available plans and their features](#)

Review + create

< Previous

Next: Networking >

Kreirajte Service Bus topic:

Nakon što kreirate Namespace, unutar njega kliknite na gumb za dodavanje novog topica na vrhu stranice. S desne strane će se otvoriti prozor unutar kojeg trebate ispuniti detalje topica. Odaberite ime topica i kliknite Create.

Home > testferbus > TestTim_rg_fer >

testferbus Service Bus Namespace

Search (Ctrl+/)

Overview

- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Settings
 - Shared access policies
 - Geo-Recovery
 - Migrate to premium
 - Encryption
 - Properties
 - Locks
- Entities
 - Queues
 - Topics
- Monitoring

Essentials

Resource group (move) : [TestTim_rg_fer](#)

Status : Active

Location : West Europe

Subscription (move) : [Syntio - Microsoft Azure \(Enterprise\)](#)

Subscription ID : a8330230-b8a0-4839-8650-17faf7ddcc42

Tags (edit) : env : dev team : labsinternal

Created : Tuesday, April

Updated : Tuesday, April

Pricing tier : [Standard](#)

Host name : testferbus.servi

Local Authentication : [Enabled](#)

Show data for the last: **1 hour** 6 hours 12 hours 1 day 7 days 30 days

Requests

Messages

Create topic

Name *

Max topic size

Message time to live Days Hours Minutes Seconds

☐ Enable auto-delete on idle topic

☐ Enable duplicate detection

☐ Enable partitioning

Create

Kreirati Service Bus subscription:

Unutar Namespace-a, u izborniku na lijevoj strani odaberite Topics i kliknite na prethodno kreirani topic. Kliknite na gumb za dodavanje novog subscriptiona na vrhu stranice. Otvara se stranica na kojoj trebate ispuniti detalje subscriptiona. Odaberite naziv i maksimalni broj pokušaja dostave poruke (Max delivery count - uobičajeni broj je 10).

Create subscription ...

Service Bus

Name * ⓘ

retailDataSubscription ✓

Max delivery count * ⓘ

10 ✓

Auto-delete after idle for ⓘ

Days

14

Hours

0

Minutes

0

Seconds

0

☐ Never auto-delete

☐ Forward messages to queue/topic ⓘ

MESSAGE SESSIONS

Service bus sessions allow ordered handling of unbounded sequences of related messages. With sessions enabled a subscription can guarantee first-in-first-out delivery of messages. [Learn more.](#)

☐ Enable sessions

MESSAGE TIME TO LIVE AND DEAD-LETTERING

Message time to live (default) ⓘ

Days

14

Hours

0

Minutes

0

Seconds

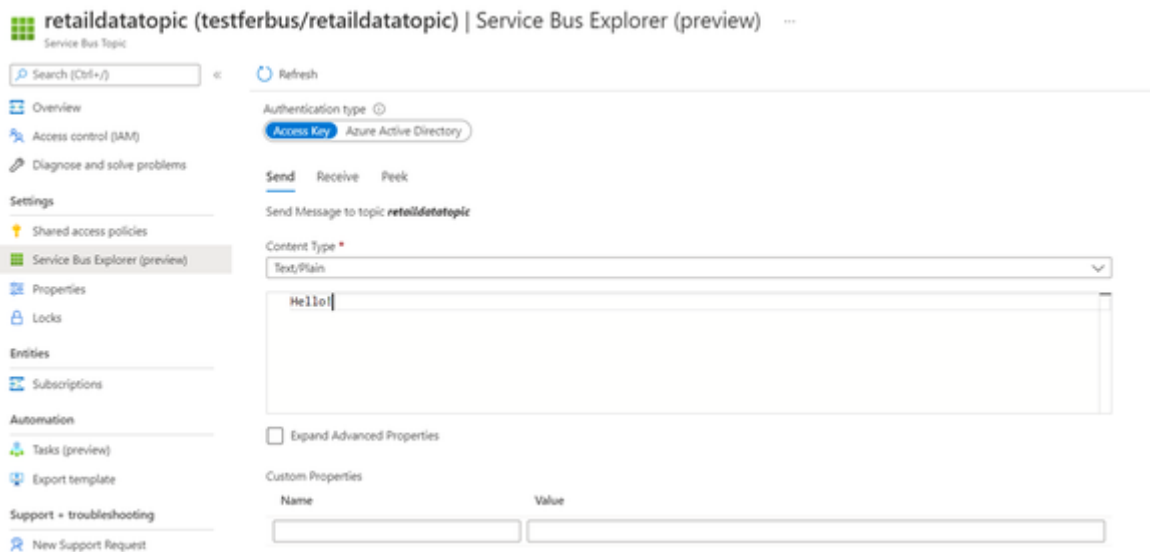
0

Create

“Ručno” slanje poruka na *Topic*

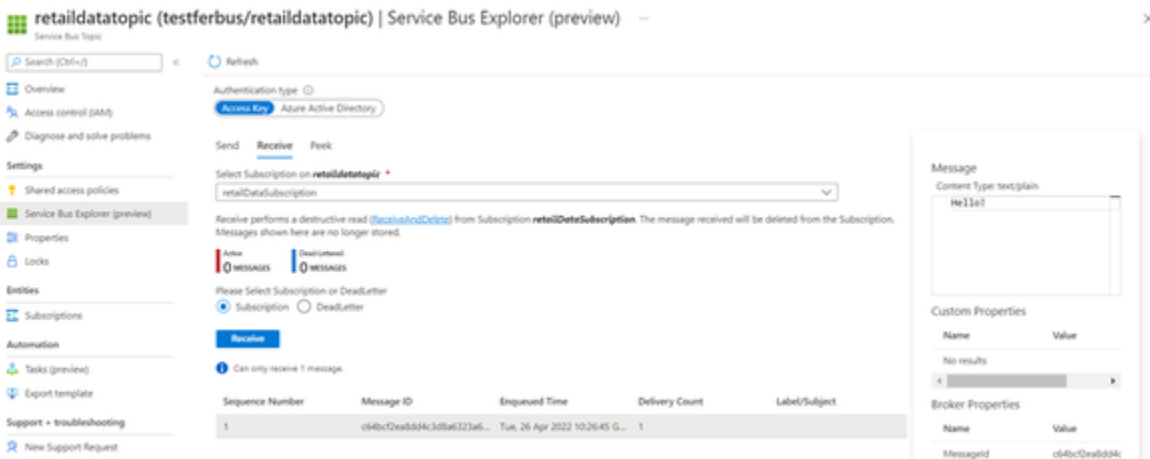
Potrebno je poslati poruku na prethodno kreirani topic pomoću Azure Portala. Nakon toga, potrebno je primiti pomoću prethodno kreiranog subscriptiona.

1. U Azure Portalu kliknite na prethodno kreirani Service Bus Namespace i odaberite Topic.
2. Otvorite Service Bus Explorer.
3. Upišite poruku po želji i kliknite Send.



4. Pod Receive izaberite prethodno kreirani subscription i kliknite Receive.

5. U tablici u donjem dijelu stranice će se pojaviti primljena poruka. Kliknite na poruku i pročitate njen sadržaj.



Slanje poruka iz funkcije na topic

- Napišite funkciju `sendOrderMessageToTopic(order Order, sender *azservicebus.Sender)` u programskom jeziku Go kojom šalžete vaše poruke na *Service Bus Topic*. Funkcija prima element tipa `Order` kojeg ste trebali kreirati u prethodnome dijelu vježbe, te `*azservicebus.Sender` koji šalže poruke na određeni *Azure Service Bus Topic*. Iskoristite `Topic` koji ste kreirali u prethodnome dijelu vježbe.
 - Koristite biblioteku: "<https://pkg.go.dev/github.com/Azure/azure-sdk-for-go/sdk/messaging/azservicebus@v0.1.0>"
- Koristeći funkcije iz prethodnog dijela zadatka, preuzmite podatke s *Data Lake-a*, pretvorite ih u odgovarajući oblik i pošaljite na *Azure Service Bus topic*
 - Connection string* za programsko spajanje na *Service Bus* pronaći ćete u Azure portalu kod pregleda *Service Bus* resursa na lokaciji: `Shared Access Policies RootManageSharedAccessKey Primary Connection String`

Napomena: pripazite da vam "topic name" bude isti u kodu kao i na vaš ranije kreirani *Service Bus Topic*.

Konzumiranje poruke sa topic-a iz funkcije

- Napišite funkciju `receiveAllOrders(client *azservicebus.Client, topicName string, subscriptionName string) []*Order` u programskom jeziku Go pomoću kojom čitate poruke iz *Service Bus* subscription-a. Funkcija prima klijenta za rad s *Azure Service Bus*, te ime topic-a i odgovarajućeg subscription-a s kojeg čita poruke. Iskoristite topic i subscription koji ste kreirali u prethodnom dijelu vježbe. Vraća sve poruke u obliku polja čiji su elementi struktura `Order`.
 - Koristite biblioteku "<https://pkg.go.dev/github.com/Azure/azure-sdk-for-go/sdk/messaging/azservicebus@v0.1.0>"
- Koristeći funkciju, konzumirajte poruke s topic-a.

Napomena: pripazite da vam "topic name" i "subscription name" u kodu odgovaraju nazivima *Service Bus* topic-a i *Service Bus* subscription-a.