

Tutorial 1a – Azure storage & compute

1. Sette opp «TweetPublisherService»
 - Tjenesten er ment å emulere en strøm av tweets. Det vil publiseres tweets fra og med 12. mai hver gang den startes. Strømmen starter på nytt ved restart. Det finnes en parameter i config som styrer raten av tweets per minutt, denne kan settes uten at tjenesten restarter.
2. Håndtere meldinger fra «tweetsqueue»
 - Køen er kommunikasjonskanalen dere skal forholde dere til.
3. Last opp håndterte tweets som «blobs», og pushe tweets videre til «searchindexqueue»
 - Denne køen er kanalen videre til neste del av systemet.
4. Provisjonere alle komponentene i Azure vha. ARM (Azure Resource Manager).

1. Sette opp TweetPublisher

- Clone prosjektet «geek-retreat-workshop» (<https://github.com/itverket/geek-retreat-workshop.git>)
- Importer solution i VS og restore nuget pakker.
- Sett start-up prosjekt til TweetPublishService og start debugging.
 - Åpne Azure Compute Emulator og se at Tweets blir publisert.
 - Åpne «View - Cloud Explorer» i VS og se at meldinger kommer inn i «tweetsqueue» under «Storage Accounts – (Development)»

2. Håndter meldinger fra kø

- Lag nytt «Azure Cloud Service»-prosjekt. («TweetHandlerService»)
 - Legg til én «Worker Role» («TweetHandler»)
 - Implementer logikk for å lese melding fra køen. Kan f.eks. logge meldingene til konsoll med Trace.
 - Set to startup prosjekter - TweetHandlerService og TweetPublisherService.

Tips1: koble til development storage med følgende logikk (OnStart()). Dette må skiftes ut senere før man eventuelt publiserer til Azure.

```
CloudStorageAccount storageAccount= CloudStorageAccount.DevelopmentStorageAccount;  
CloudQueueClient queueClient = storageAccount.CreateCloudQueueClient();  
CloudQueue tweetsQueue = queueClient.GetQueueReference("tweetsqueue");  
tweetsQueue.CreateIfNotExists();
```

Tips2: Les meldinger fra køen og husk å slette etter håndteringen er fullført (RunAsync()).

```
var msg = await tweetsQueue.GetMessageAsync(cancellationToken);  
await tweetsQueue.DeleteMessageAsync(msg, cancellationToken);
```

3. Last opp meldingene fra køen som blobs.

- Koble nå til en ny «blob container» («handledtweets»)
- Debug systemet og se at meldingene blir håndtert og lastet opp som blobs.

Tips3: Bruk samme storage account som i forrige steg, men lag nå en blob client.

```
CloudBlobClient blobClient = storageAccount.CreateCloudBlobClient();
CloudBlobContainer blobContainer =
    blobClient.GetContainerReference("handledtweets");
blobContainer.CreateIfNotExists();
```

Tips4: Blobs kan lastes opp på mange ulike måter avhengig av innhold, den enkleste er så en enkel string.

```
var blob = blobContainer.GetBlockBlobReference(msg.Id);
await blob.UploadTextAsync(msg.AsString, cancellationToken)
```

4. Push meldingene til searchindexqueue.

- Koble nå til en ny kø («handledtweets»)
- Denne gangen er det ikke nødvendig å lage ny klient, bruk den samme som i steg 2.
- Debug og sjekk at meldinger nå blir håndtert, lastet opp til blob og pushet til den nye køen.

Tips5: Bruk samme storage account og «queue client» som tidligere.

```
CloudQueue searchQueue = queueClient.GetQueueReference("searchindexqueue");
searchQueue.CreateIfNotExists();
```

```
searchQueue.AddMessage(new CloudQueueMessage(msg.AsString));
```

Tutorial 1b – Resource Manager Provisioning

5. Provisjoner Azure-resurser for hele systemet

- Sørg for å ha Azure Powershell modulene installert og klare til bruk.
(<https://azure.microsoft.com/en-us/documentation/articles/powershell-install-configure/>)
- Koble til «Azure subscription», bruk enten personlig konto med MSDN, eller «ITverket playground» subscriptionen. Bruk de 3 følgende commandletsene.

1. Logg inn med «work account» eller Microsoft account, vha. Add-AzureAccount (Azure AD)

```
λ Add-AzureAccount
```

Id	Type	Subscriptions
anders.kofoed@outlook.com	User	2b60a02f-805b-49e9-812a-04304091867b 8c006a3c-f5b0-4b23-a99c-bfc2fb741be1

2. List alle tilgjengelige Azure subscriptions med Get-AzureSubscription

```
λ Get-AzureSubscription | select SubscriptionId, SubscriptionName | Format-List
```

SubscriptionId : 8c006a3c-f5b0-4b23-a99c-bfc2fb741be1
SubscriptionName : Visual Studio Enterprise med MSDN

SubscriptionId : 2b60a02f-805b-49e9-812a-04304091867b
SubscriptionName : Windows Azure Cloud Essentials

SubscriptionId : dca72209-8a25-4ce3-83e3-51ec04e5ac2a
SubscriptionName : ITverket Drift Pay-As-You-Go

SubscriptionId : 00f3b613-a2d4-4828-97f4-a28d906969c0
SubscriptionName : ITverket Azure Playground Pay-As-You-Go

3. Velg en subscription med Select-AzureSubscription

```
λ Select-AzureSubscription -SubscriptionId 8c006a3c-f5b0-4b23-a99c-bfc2fb741be1
```

- Kjør scriptet <repos-dir>/geek-retreat-workshop/AzureManagement/Create-GRResourceGroup.ps1
- Velg et kort gruppenavn eller bruk gruppenummer (må være kort grunnet begrensninger på «storage account» navn)
- Følg med at alt går greit, og sjekk gjerne i portalen at resursene har blitt provisjonert som ønsket. ☺