Tutorial 1 – Azure storage & compute

Sette opp Tweetpubliserer

- Clone prosjektet GeekRetreatInit (https://ksitv.visualstudio.com/DefaultCollection/GeekRetreat2015/_git/GeekRetreatInit)
- Importer sollution in VS og restore nuget pakker.
- Sett startup prosjekt til TweetPublishService og start debugging.
    - Åpne Azure Compute Emulator og se at Tweets blir publisert.
    - Åpne Cloud Explorer i VS og se at meldinger kommer inn i «tweetsqueue»

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.

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

```
CloudStorageAccount storageAccount= CloudStorageAccount.DevelopmentStorageAccount;
CloudQueueClient queueClient = _storageAccount.CreateCloudQueueClient();
CloudQueue cloudQueue = _queueClient.GetQueueReference("tweetsqueue");
_cloudQueue.CreateIfNotExists();
```

Tips2: Les meldinger fra køen og husk å slette etter håndteringen er fulført

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

Last opp meldingene fra køen som blobs.

- Koble nå til en ny «blob container» («handledtweets»)
- Set to startup prosjekter - TweetHandlerService og TweetPublisherService.
- 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();
```