

Azure Function / Azure  
Storage

# Azure Function Apps

- Det är små kod-delar eller funktioner som körs i molnet.
- Du kan skriva kod utan att oroa dig för hela din app eller infrastruktur.
- Detta är också en så kallad "Extension" av Azure App service



# Azure Function Apps - Språk

- Språk som stöds av funtions apps:
- C# .NET & .NET Core
- Java
- JavaScript (NodeJs)
- Python
- F#
- PowerShell
- TypeScript



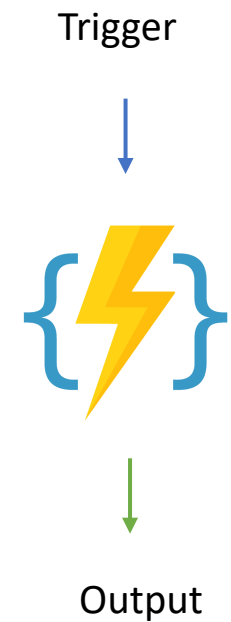
# Azure Function Apps - Pris

- "Pay-per-use" pris (Billigt)
- Du betalar bara för det du använder.
- Du kan också använda "dependencies" (NPM & NuGet)
- Integrerad säkerhet.
- Enkelt att integrera
- Du kan utveckla genom portalen eller VS code samt VS studio.
- Integrering med Github, Azure DevOps osv..
- Hela SDK:t är Open-Source. (Finns på Github)



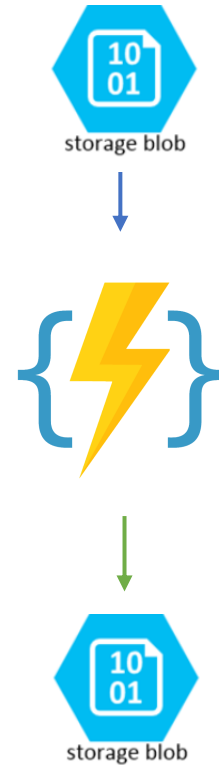
# Hur funkar detta?

- Function är en kod.
- Men för att exekvera koden måste du ha en "trigger"
- En trigger är en "action" som skickar info så den kan trigga kod att köra.
- Och när den är klar så får vi en output.



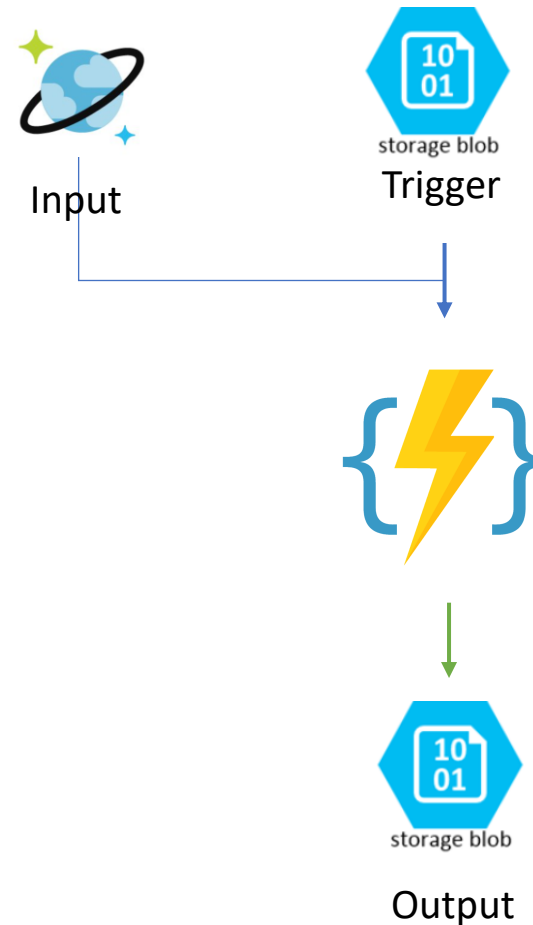
# Hur funkar detta?

- Du kan t.ex trigga en function när det kommer in en specifik fil.
- Och sedan kan du "skicka" tillbaka filen till en blob storage när det är färdigt.



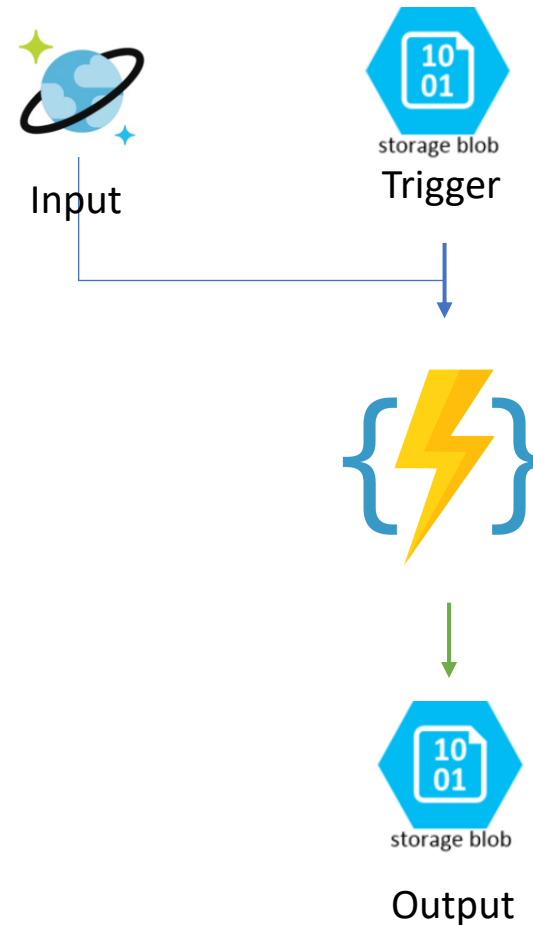
# Hur funkar detta?

- Vi kan bara ha 1 Trigger men vi kan ha flera inputs om vi vill.
- Trigga function när fil kommer in.
- Hämta info från Databas (CosmosDB)
- Processa
- Output



# Hur funkar detta?

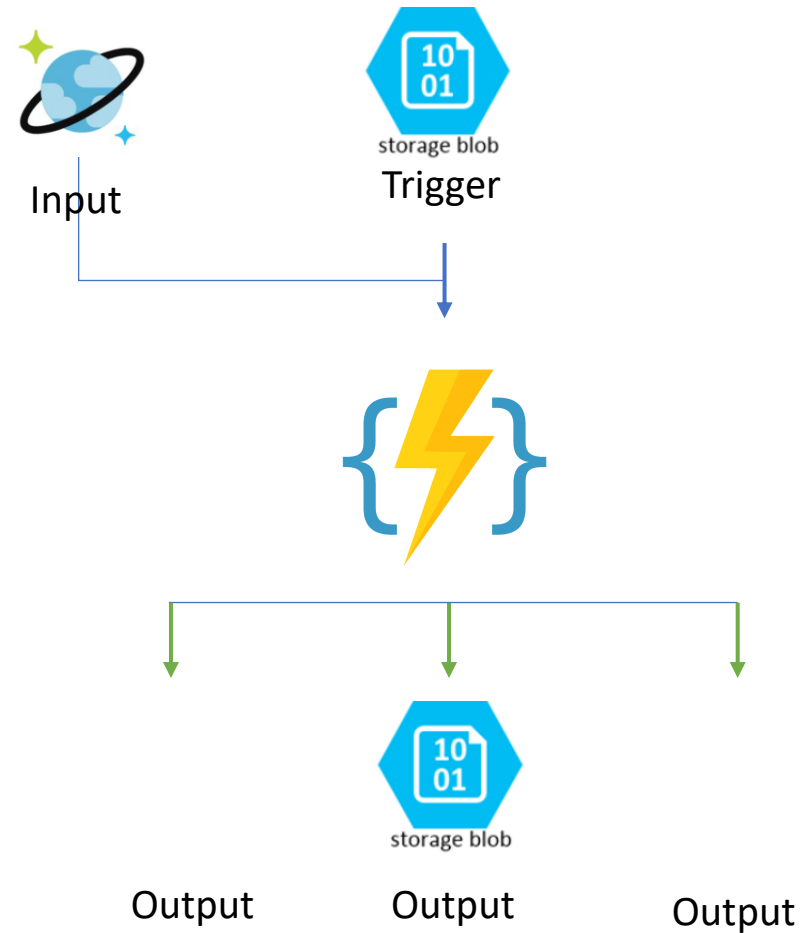
- Vi kan bara ha 1 Trigger men vi kan ha flera inputs om vi vill.
- Trigga function när fil kommer in.
- Hämta info från Databas (CosmosDB)
- Processa
- Output





# Hur funkar detta?

- Vi kan också ha flera outputs



# Azure Storage

- Spara stora filer, bilder etc.
- SQL
- NoSQL



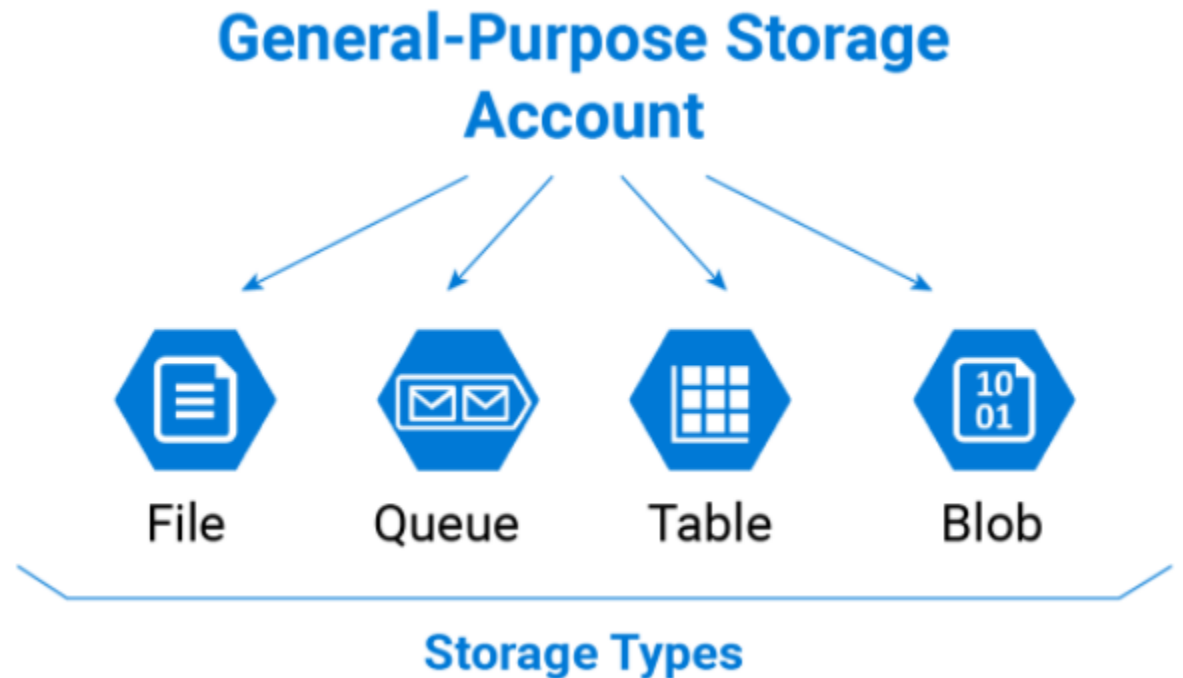
# Azure Storage

- Fördelar med **Azure storage**:
- Hållbarhet – Microsoft tar hand om alla hårdvarufel
- Säkerhet – Din data är krypterad
- Skalbarhet – Du kan ha upp till Petabytes av data
- Tillgänglighet - Finns bra API:er och SDK:er



# Azure Storage

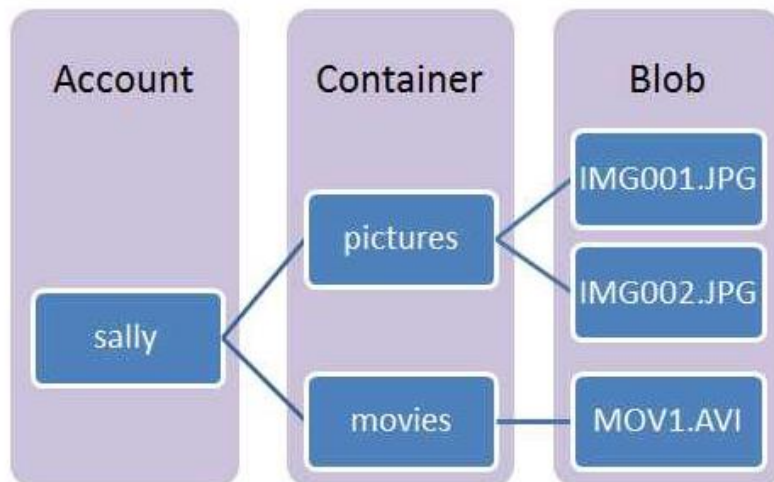
- Finns 4 olika tjänster i en **storage account**:
- **Blob storage**
- **File storage**
- **Storage Queue**
- **Table Storage**



# Blob storage

- En massiv och skalbar lagring för text och binär data.
- Med andra ord vilken fil som helst går att lagra

Struktur:



Blob

# Blob storage – Åtkomstnivåer

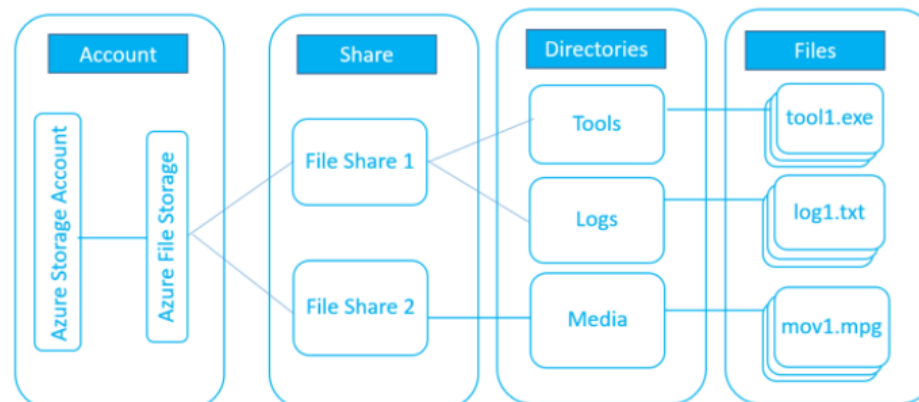
- **Hot** - Optimerad för frekvent åtkomst av objekt
- **Cool** - Optimerad för att lagra stora mängder data som sällan nås och lagras i minst 30 dagar
- **Archive** – Optimerad för data som kan ha en längre "latency" när du ska hämta data och kommer dessutom att arkiveras i minst 180 dagar.



Blob

# File Storage

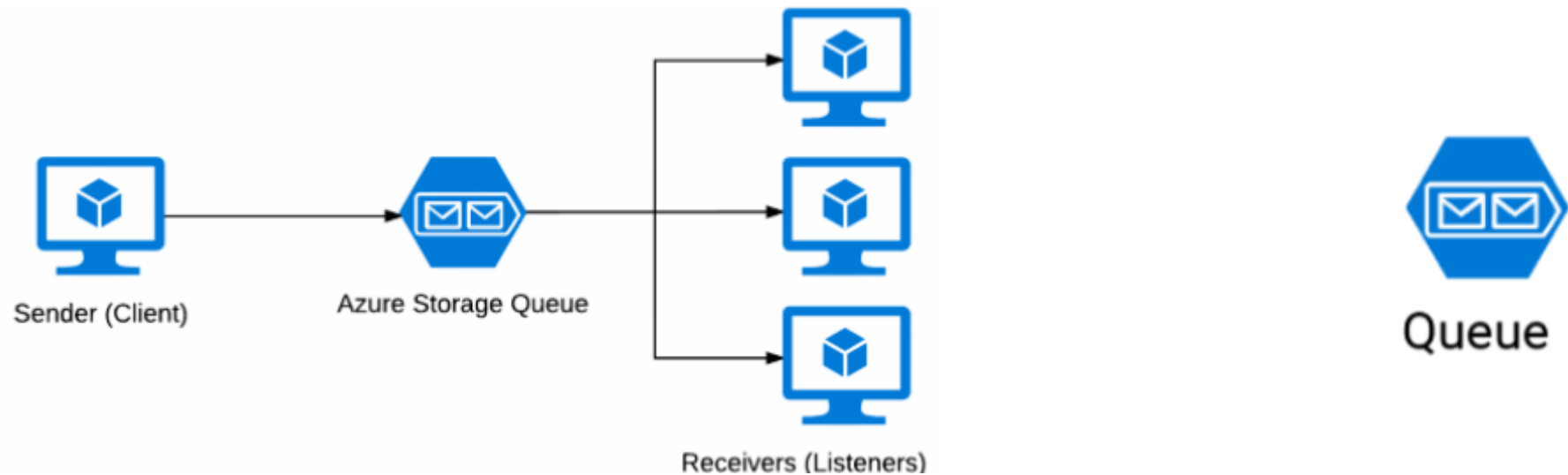
- Principen är densamma som blob storage men istället för containers
- Så har man något som kallas för "Share" och i varje "Share" har du filer och mappar.
- Medans i en blob storage kan du bara lagra filer.



File

# Storage Queue

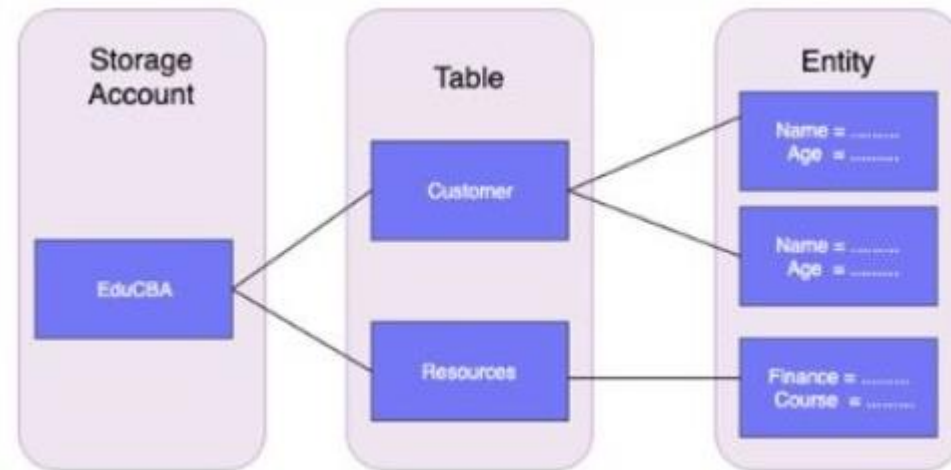
- Används för hållbar meddelande hantering mellan applikationer t.ex.





# Table Storage

- En NoSQL schemalös lagring för strukturerad data.

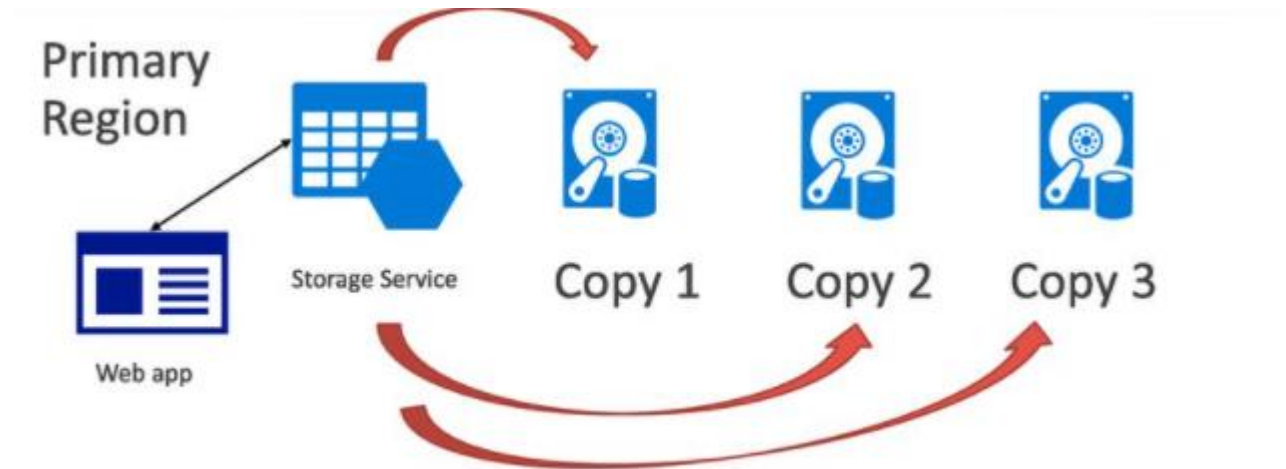


# Data redundancy

- Locally-redundant storage (**LRS**)
  - Zone-redundant storage (**ZRS**)
  - Geo-redundant storage (**GRS**)
  - Read-access geo-redundant storage (**RA-GRS**)
- 
- Detta är tjänster som låter dig bestämma hållbarhet av din data

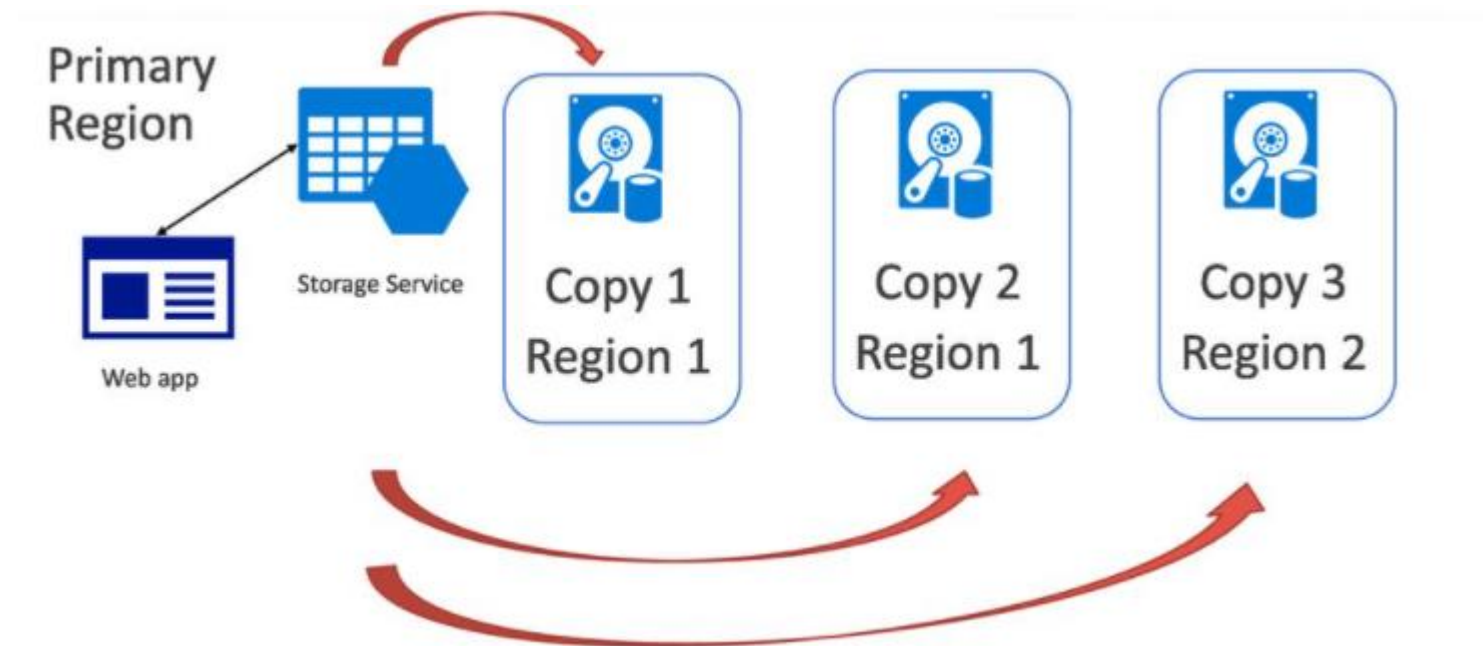
# Locally-redundant storage (LRS)

- Här kopieras din data i 1 region i 3 olika diskar.
- Även kallat **scale unit**



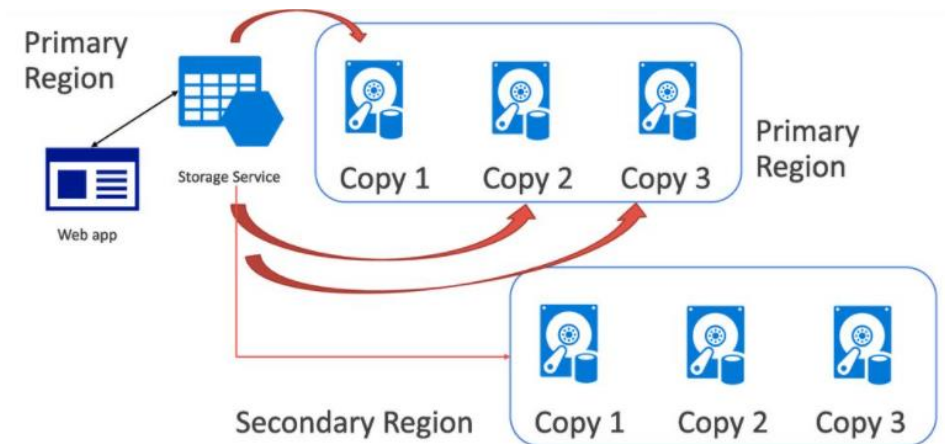
# Zone-redundant storage (ZRS)

- Denna kostar lite mer.
- Men detta betyder att varje kopia har:
  - Separat strömkälla
  - Separat kylning
  - Separat nätverk
- I detta fall har du då 3 scale units



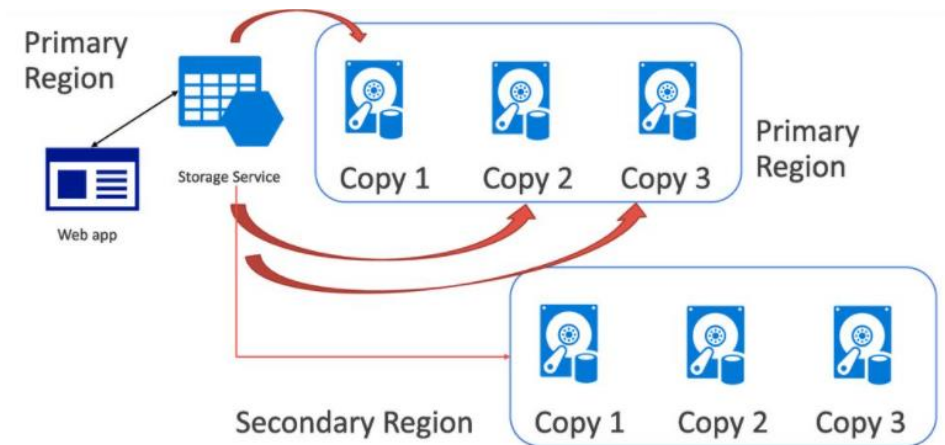
# Geo-redundant storage (GRS)

- Detta är som att du har 2 stycken LRS i olika regioner.
- Varje LRS har en egen scale unit
- Men du får bara läsa/skriva data från **Primary Region** men inte från **Secondary Region**



# Read-access geo-redundant storage (RA-GRS)

- Detta är precis som GRS men här får du kan få läsa data från Secondary Region om så önskas.



# Veckans Labbar

- Börja med att titta på inspelning.
  - Gör sedan labb 1 & 2
  - I labb 3 ska du skicka och ta emot fil med hjälp av blob. Instruktioner finns i README.
- 
- 1. Azure Function HTTP trigger: [Länk](#)
  - 2. Azure Function triggered by blob: [Länk](#)
  - 3. Finns även en labb i teams för blob storage.