# Orleans Silo正式環境部署注意事項

Orleans 在 Silo 部署(Ops)相關有一些套件以及API，可以協助我們在正式環境部署時，讓系統更穩定，以下介紹這些套件以及API。

## Grain Directory

Orleans的Grain在其生命週期啟動後，會跟Silo叢集註冊一個當時的唯一識別ID，以便讓後續的RPC呼叫進來時，才會知道要將RPC呼叫導向到哪個Silo上的Grain實體。這些Grain的註冊資訊，就是放在稱為 “Grain Directory” 的Orleans框架底層機制來儲存的。

Orleans的Silo預設會使用In-Memory的機制以分散式雜湊表(Distributed Hash table)將資料儲存於每台Silo伺服器上，使用Orleans框架自身研發的同步演算法來達到狀態的最終一致性同步；

這樣的機制在開發時期沒有什麼問題，但是在正式環境部署時，為了避免當單一節點故障，導致Grain Directory的資料還要在每台伺服器上重新同步，Orleans在 v3.2 之後提供外部儲存機制，讓Grain Directory可以儲存在外部的Azure Table或是Redis這兩種No-SQL儲存資料庫之中，使得Silo要關掉/重啟的速度可以更快，而且不會有兩個相同的Grain突然在多個Silo上由於Grain Directory暫時不同步而同時在多台伺服器上啟動的靈異現象。

要使用Grain Directory的外部儲存機制，需要在欲使用的Grain實作宣告上加入 [[GrainDirectory]](https://learn.microsoft.com/en-us/dotnet/api/orleans.graindirectory.graindirectoryattribute) 的屬性，並指定其 [GrainDirectoryName](https://learn.microsoft.com/en-us/dotnet/api/orleans.graindirectory.graindirectoryattribute.graindirectoryname) 的成員屬性值：

[GrainDirectory(GrainDirectoryName= "my-grain-directory")]  
public class MyGrain : Grain, IMyGrain  
{  
 // ...  
}

如上的範例，MyGrain 設定使用的 *GrainDirectoryName* 為 my-grain-directory；而在Silo的配置，使用Azure Table就安裝 [Microsoft.Orleans.GrainDirectory.AzureStorage](https://www.nuget.org/packages/Microsoft.Orleans.GrainDirectory.AzureStorage) Nuget套件，在配置程式碼的寫法仿效配置Grain State的設計，呼叫 [AddAzureTableGrainDirectory()](https://learn.microsoft.com/en-us/dotnet/api/orleans.hosting.azuretablegraindirectoryextensions.addazuretablegraindirectory) 這個擴充方法並在第一個參數填入前面Grain設定的GrainDirectoryName成員屬性的字串值：

siloBuilder.AddAzureTableGrainDirectory(  
 "my-grain-directory",  
 options => options.ConnectionString = "Actual Azure Table Connection string ... ");

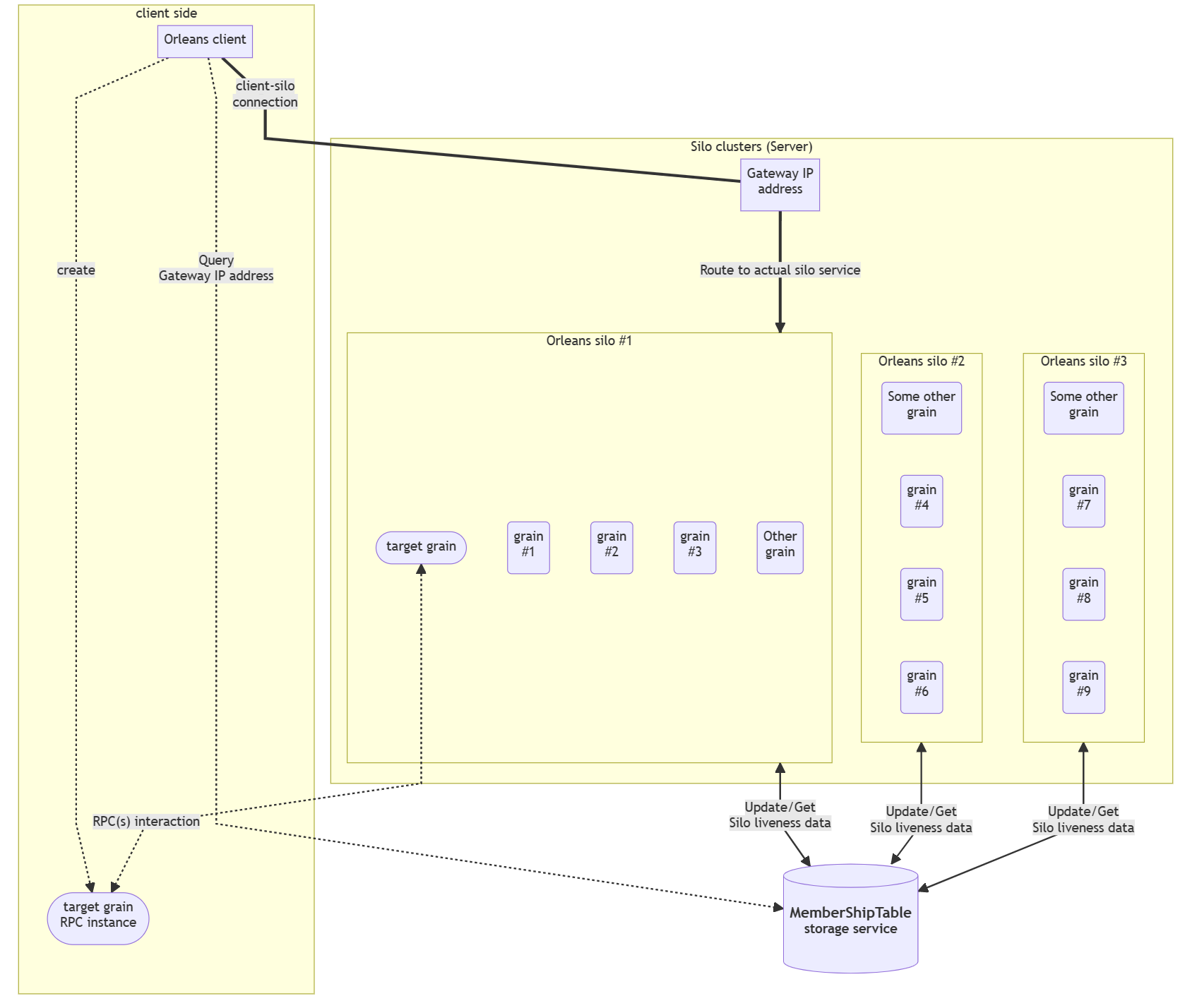
使用Redis就安裝 [Microsoft.Orleans.GrainDirectory.Redis](https://www.nuget.org/packages/Microsoft.Orleans.GrainDirectory.Redis) Nuget套件，在配置程式碼的寫法仿效配置Grain State的設計，呼叫 [AddRedisGrainDirectory()](https://learn.microsoft.com/en-us/dotnet/api/orleans.hosting.redisgraindirectoryextensions.addredisgraindirectory) 這個擴充方法並在第一個參數填入前面Grain設定的GrainDirectoryName成員屬性的字串值：

siloBuilder.AddRedisGrainDirectory(  
 "my-grain-directory",  
 options => options.ConfigurationOptions = new ConfigurationOptions  
 {  
 EndPoints = { "Actual Redis address" },  
 Password = "Actual Redis password ... "  
 });

以上這兩種都有提供 [UseAzureTableGrainDirectoryAsDefault()](https://learn.microsoft.com/dotnet/api/orleans.hosting.azuretablegraindirectoryextensions.useazuretablegraindirectoryasdefault) 和 [UseRedisGrainDirectoryAsDefault()](https://learn.microsoft.com/dotnet/api/orleans.hosting.redisgraindirectoryextensions.useredisgraindirectoryasdefault) 的擴充方法，以便將其提供的GrainDirectory儲存機制設定為預設的，就不用為一一替每個Grain的GrainDirectoryName做配置。

## Cluster Membership 的連線微調

在運營環境的多台Silo之間的溝通，以及Client如何得知Silo的位址以便進行RPC呼叫，都是由儲存在MembershipTable中的Silo資訊來得知的，如下圖所示右下角的Membership storage service：



所以像是使用 Azure Table, ADO.NET 的 Cluster Provider，Client端就必須要有能夠連線到這些資料庫的權限，而且這些資料庫的連線也必須要設定在Client端的ClientBuilder配置程式碼中，才能夠正常的運作。

在使用單台伺服器跑單一Silo（在SiloBuilder配置程式碼使用 [UseLocalhostClustering()](https://learn.microsoft.com/dotnet/api/orleans.hosting.corehostingextensions.uselocalhostclustering) 擴充方法）時不需要什麼額外套件安裝和設定，但若需要水平擴展(Scale-Out)跑多台Silo跨越多台伺服器共同服務時，就必須要安裝並設定相關Clustering Provider的Nuget套件，才能正常運作；目前官方有提供下列幾種Clustering Provider：

* [Microsoft.Orleans.Clustering.AzureStorage](https://www.nuget.org/packages/Microsoft.Orleans.Clustering.AzureStorage)：使用 Azure Table Storage 作為Cluster Membership的儲存機制。
* [Microsoft.Orleans.Clustering.AdoNet](https://www.nuget.org/packages/Microsoft.Orleans.Clustering.AdoNet)：跟Storage Provider一樣，是底層使用 ADO.NET 來連結SQL-based資料庫來作為Cluster Membership，官方提供[MS-SQL, MySQL, Oracle PostgreSQL四種資料庫資料表建立Script](https://learn.microsoft.com/en-us/dotnet/orleans/host/configuration-guide/adonet-configuration#clustering)，也是跟Storage Provider一樣的順序：先執行Main scripts，然後再執行Clustering script的方式來建立。
* [Microsoft.Orleans.Clustering.Consul](https://www.nuget.org/packages/Microsoft.Orleans.Clustering.Consul)：使用 [Consul](http://consul.io) 這個開源的[分散式組態儲存/服務發現(Service Discovery)解決方案](https://columns.chicken-house.net/2017/12/31/microservice9-servicediscovery/#%E6%A1%88%E4%BE%8B-consul)作為Cluster Membership的儲存機制。
* [Microsoft.Orleans.Clustering.ZooKeeper](https://www.nuget.org/packages/Microsoft.Orleans.Clustering.ZooKeeper)：使用 [Apache ZooKeeper](https://zookeeper.apache.org/) 這個開源的分散式組態儲存服務作為Cluster Membership的儲存機制。

第三方貢獻的開源Provider中，也有像是 [Orleans.Providers.MongoDB](https://github.com/OrleansContrib/Orleans.Providers.MongoDB) 是使用MongoDB，以及 [Orleans.Clustering.Redis](https://github.com/OrleansContrib/Orleans.Redis#orleansclusteringredis) 使用Redis來儲存以及同步Cluster Membership資訊。

在寫SiloBuilder配置程式碼時，可以藉由設定 [ClusterMembershipOptions](https://learn.microsoft.com/dotnet/api/orleans.configuration.clustermembershipoptions) 這個屬性來微調一些Cluster的資訊同步動作Timeout時間，以便克服網路通訊不穩定的狀況，如下範例：

siloBuilder.Configure<ClusterMembershipOptions>(options =>  
{  
 options.ProbeTimeout= TimeSpan.FromSeconds(30);  
 options.TableRefreshTimeout = TimeSpan.FromSeconds(300);  
});

### Silo 服務的IP位址和Port設定

在SiloBuilder配置程式碼可以透過設定 [EndPointOptions](https://learn.microsoft.com/dotnet/api/orleans.configuration.endpointoptions) 來設定Silo服務的IP位址和Port，如下範例：

siloBuilder.Configure<EndpointOptions>(options =>  
{  
 options.AdvertisedIPAddress = IPAddress.Parse(siloIpString);  
 options.SiloPort = siloPort;  
 options.GatewayPort = gatewayPort;  
});

也可以使用另一個SiloBuilder的擴充方法 [ConfigureEndpoints()](https://learn.microsoft.com/dotnet/api/orleans.hosting.endpointoptionsextensions.configureendpoints) 的擴充方法來設定。

上述兩種方法都可以設定Silo服務的IP位址和Port，但 ConfigureEndpoints() 還有其他可以設定hostname, IPv4或v6，以及是否使用主機上的所有IP位址(listenOnAnyHostAddress)的重載(overload)方法及控制參數。

## .NET CLR Garbage Collection 設定

跑Silo的 .NET 程式建議要啟用[Server GC](https://learn.microsoft.com/dotnet/standard/garbage-collection/workstation-server-gc#server-gc)，以便在Silo上的Grain有更好的執行效能，啟用的方法有兩種：

* 編譯時期(Build time)的Silo專案檔設定：可從 Silo 專案的 .csproj 檔案中，將 ServerGarbageCollection 屬性設定為 true，如下範例：
* <Project Sdk="...">  
   <PropertyGroup>  
   <!-- Other settings -->  
   <ServerGarbageCollection>true</ServerGarbageCollection>  
   </PropertyGroup>  
  </Project>
* 執行時期(Run time)設定：在執行前設定環境變數 DOTNET\_gcServer 為數字 ‘1’，或是修改以dotnet publish 指令輸出部署目錄中的 *[appname].runtimeconfig.json* ，增加 configProperties 的 System.GC.Server 屬性，如下範例：
* {  
   "runtimeOptions": {  
   // ...  
   "configProperties": {  
   "System.GC.Server": true  
   // Other runtime settings  
   }  
   }  
  }

明天來介紹Orleans應用在Azure雲端使用Azure Web Apps的部署方式。