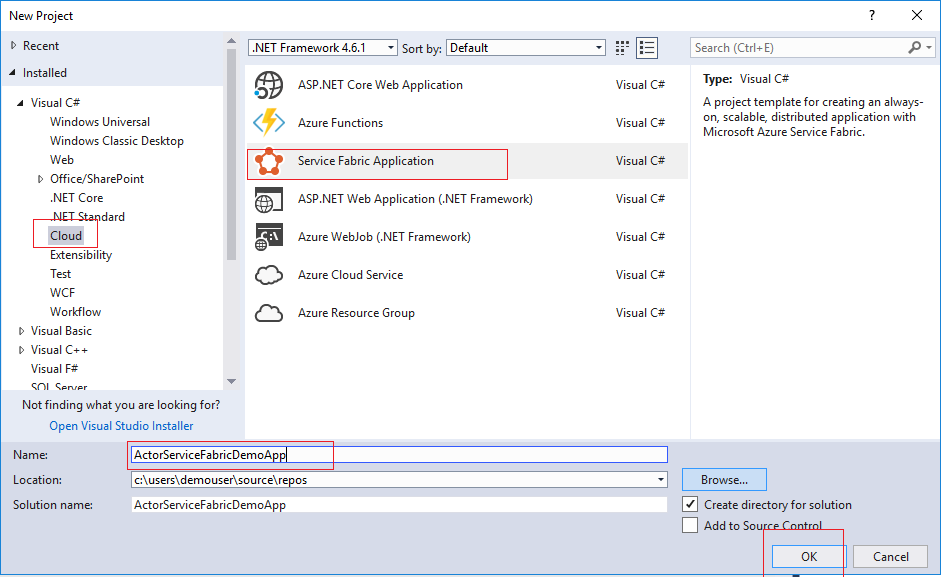
**Service Fabric Actor Step-by-Step**

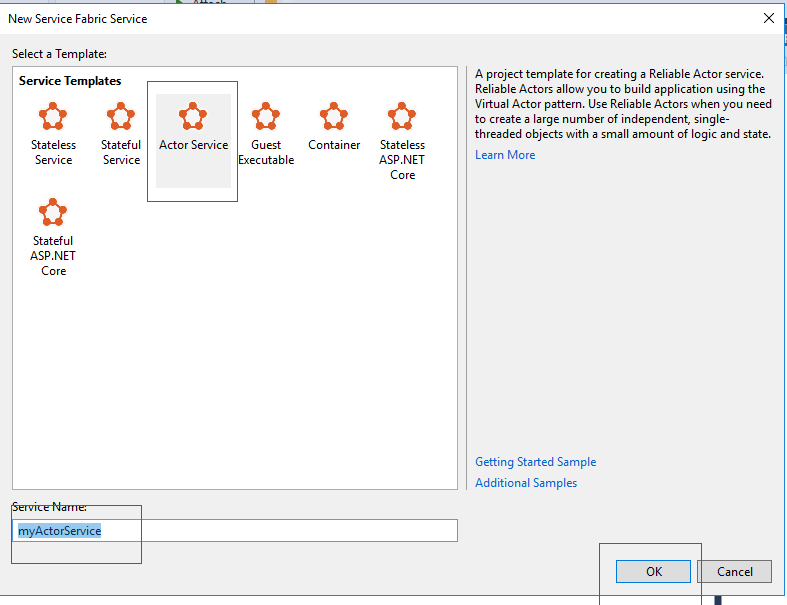
Agenda : in this Tutorial, we will create a simple Service Fabric Actor that will give us the hello world string. Then after we will create a Actor client using which we will call the actor and will get the hello world string.

Step: Create a new project by clicking : File + New + Project

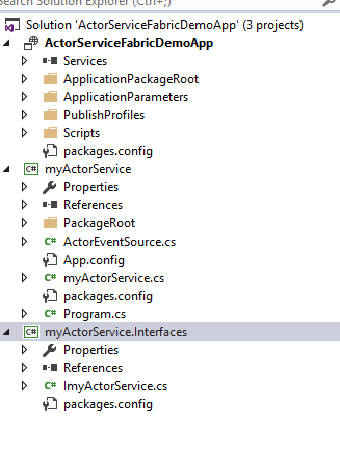
Then Click on Cloud and select Service Fabric Appliclation and give the name “ActorServiceFabricDemoApp” to the application and click on Ok button.



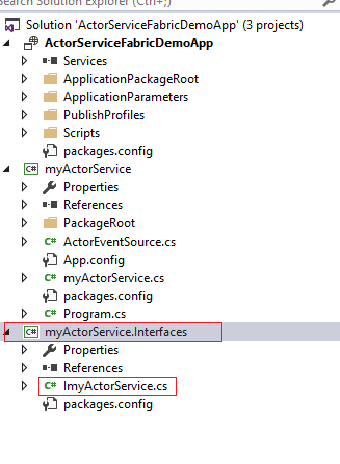
On the next screen, Select Actor Service and give it a name ActorService and then click on ok button.



The created project shows the following structure:



In the myActorService.Interfaces project, in the ImyActorService.cs file, replace the interface definition as follows:

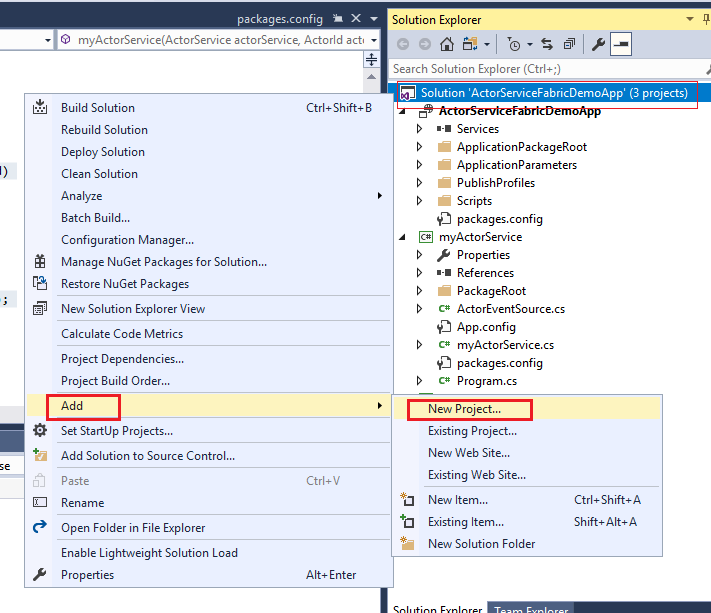


|  |
| --- |
| public interface ImyActorService : IActor  {    Task<String> GetHelloWorldAsync();    } |

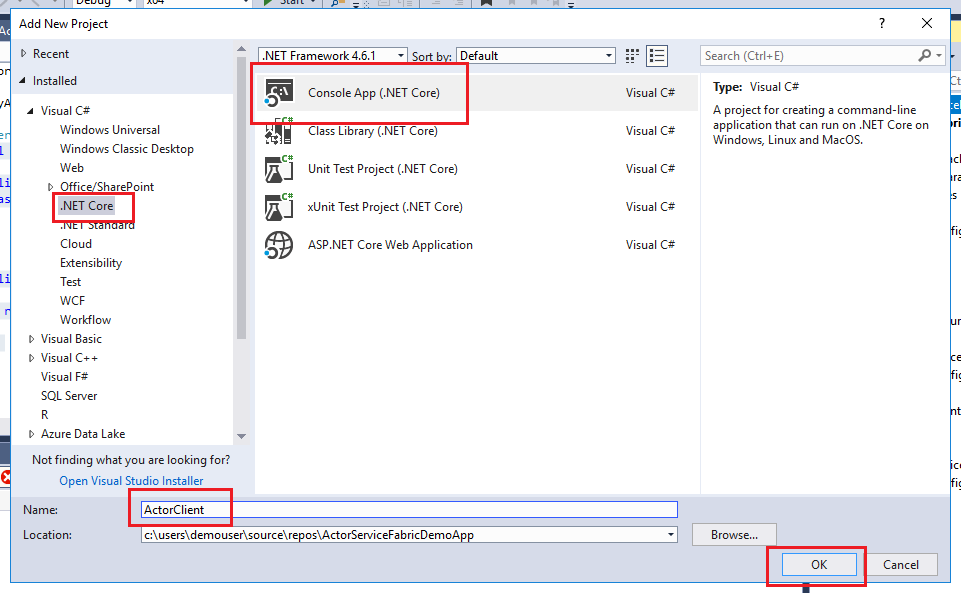
In the myActorService project, in myActorService**.cs**, replace the entire class definition as follows:

|  |
| --- |
| internal class myActorService : Actor, ImyActorService  {  public myActorService(ActorService actorService, ActorId actorId)  : base(actorService, actorId)  {  }  public Task<String> GetHelloWorldAsync()  {  return Task.FromResult("Hello World from my reliable Actor");  }    } |

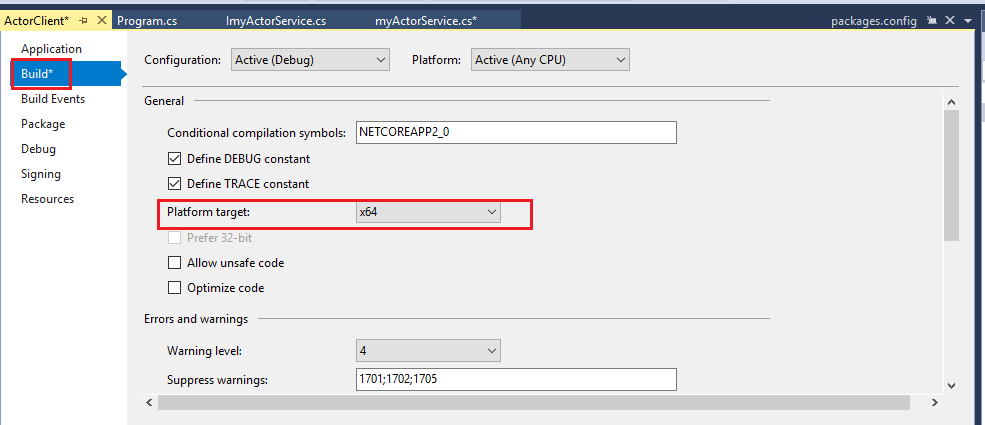
Now r ight-click on the solution in Solution Explorer > **Add** > **New Project..**

****

Under the **.NET Core** project types, choose **Console App (.NET Core)**. Name the project ActorClient.

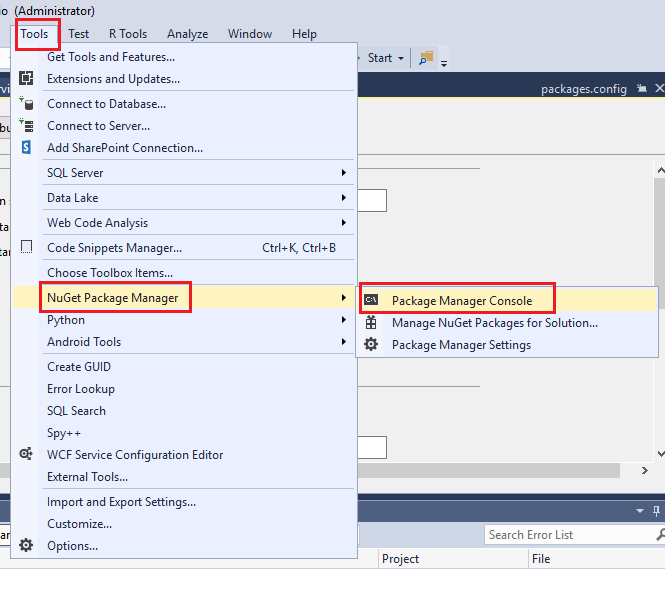


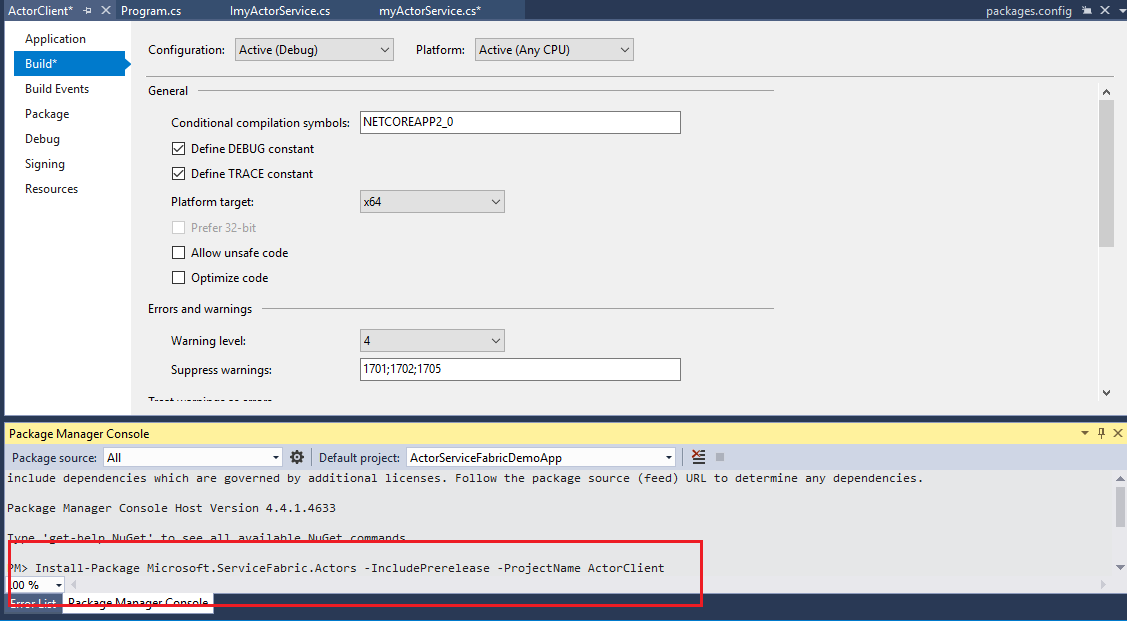
In Solution Explorer, right-click the **ActorClient** project, and then click **Properties**. On the **Build** tab, set **Platform target** to **x64**.

****

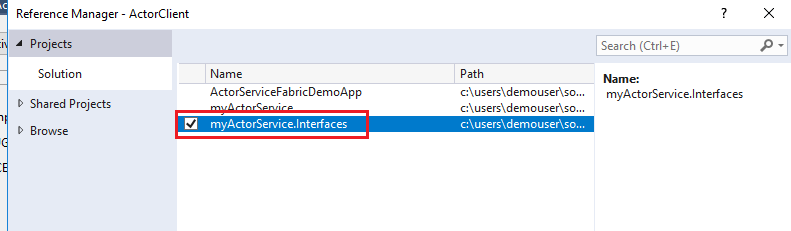
Click **Tools** > **NuGet Package Manager** > **Package Manager Console**. In the Package Manager Console, enter the following command:

|  |
| --- |
| Install-Package Microsoft.ServiceFabric.Actors -IncludePrerelease -ProjectName ActorClient |





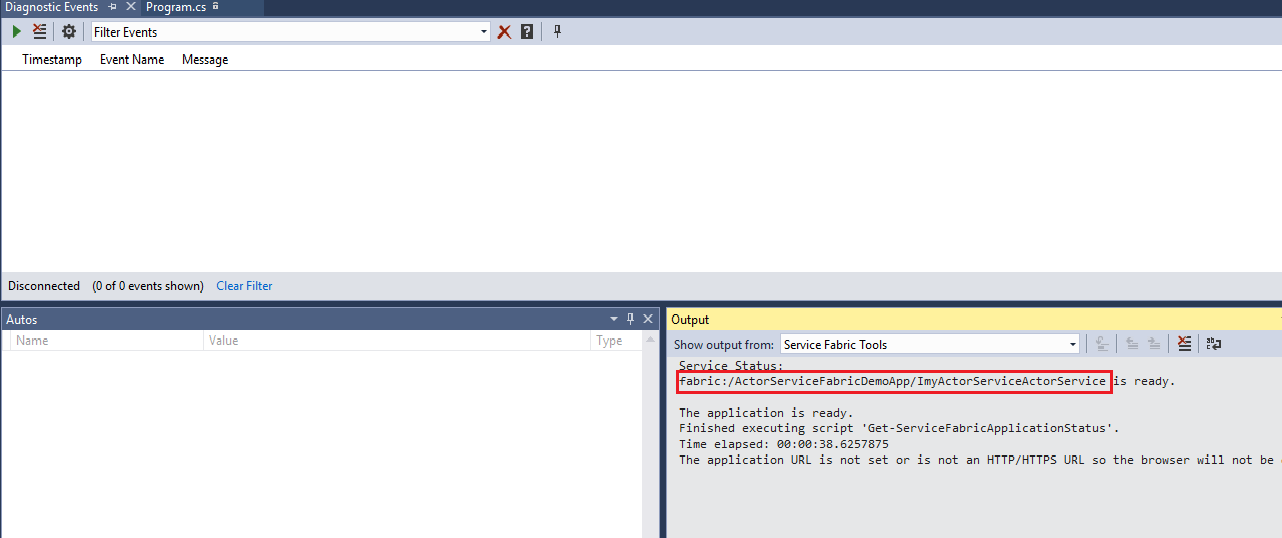
In the ActorClient project, right-click **Dependencies** and then click **Add reference...**. Select **Projects > Solution** (if not already selected), and then tick the checkbox next to **myActorService.Interfaces**. Click **OK**.



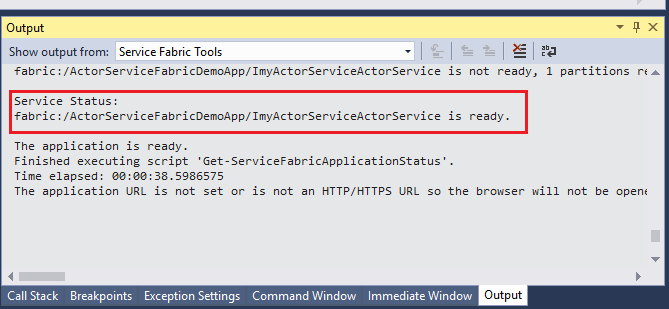
In the ActorClient project, replace the entire contents of Program.cs with the following code:

|  |
| --- |
| static void Main(string[] args)  {  ImyActorService actor = ActorProxy.Create<ImyActorService>(ActorId.CreateRandom(),  new Uri("fabric:/ActorServiceFabricDemoApp/ImyActorServiceActorService"));  Task<string> retval = actor.GetHelloWorldAsync();  Console.Write(retval.Result);  Console.ReadLine();  } |

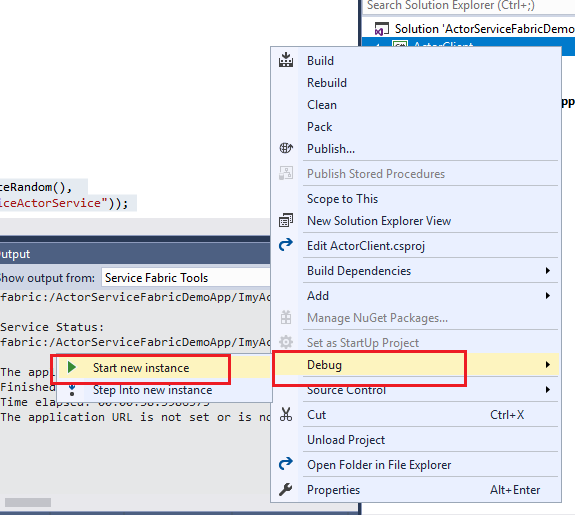
Note : you will find the actor service uri when you run the service by pressing F5 in the output window.

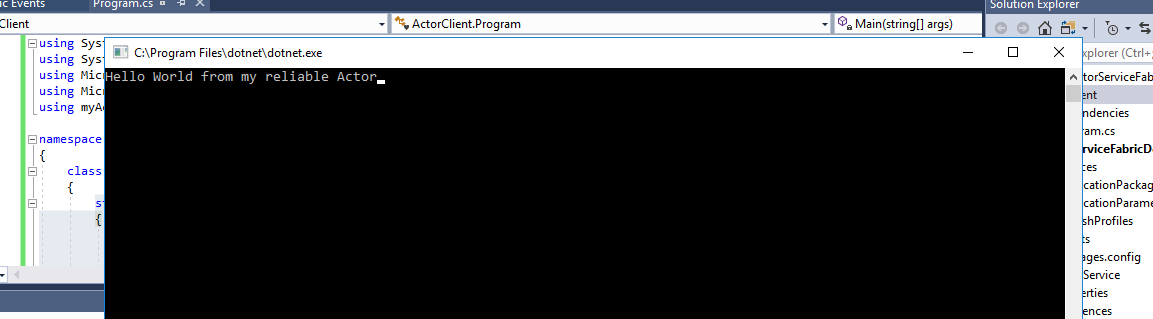


Press **F5** to build, deploy, and run the application locally in the Service Fabric development cluster. During the deployment process, you can see the progress in the **Output** window.



While the service is in running state, In Solution Explorer, right-click on the **ActorClient** project, then click **Debug** > **Start new instance**. The command line application should display the output from the actor service.





Done. You have successfully created a actor service and in the client application you have used it and get the hello world message.