Isaac Lim
2/10/18

McKesson | Final Project Service Fabric | Deep Azure

PROBLEM STATEMENT

A Company is wanting to expand their infrastructure, and one of the initiatives is to create a scalable service fabric for their client facing web service. Create a service fabric cluster containing 1 node and deploy a web service application which will return a row from the dataset.

OVERVIEW

Microsoft Azure Service Fabric gives the user the ability to easily package and deploy applications and services. As well, Service Fabric clusters provide high scalability and reliability for these applications and services. Service Fabric is highly customizable. For instance, you can create a cluster that has multiple VMs each being a node, or you can have a single VM with multiple nodes attached. Applications can be deployed quickly once a Service Fabric cluster is created, unlike traditional deployments where a VM needs to be created every time.

STEPS

1. Install Service Fabric SDK
2. Create Service Fabric Cluster
3. Create sample application
4. Deploy application to Service Fabric Cluster
5. Run application web service

DATA

https://catalog.data.gov/dataset/age-adjusted-death-rates-for-the-top-10-leading-causes-of-death-united-states-2013

HARDWARE

Windows 7 64 bit on Core I5-6300HQ

SOFTWARE

- Service Fabric SDK
- Microsoft Visual Studio

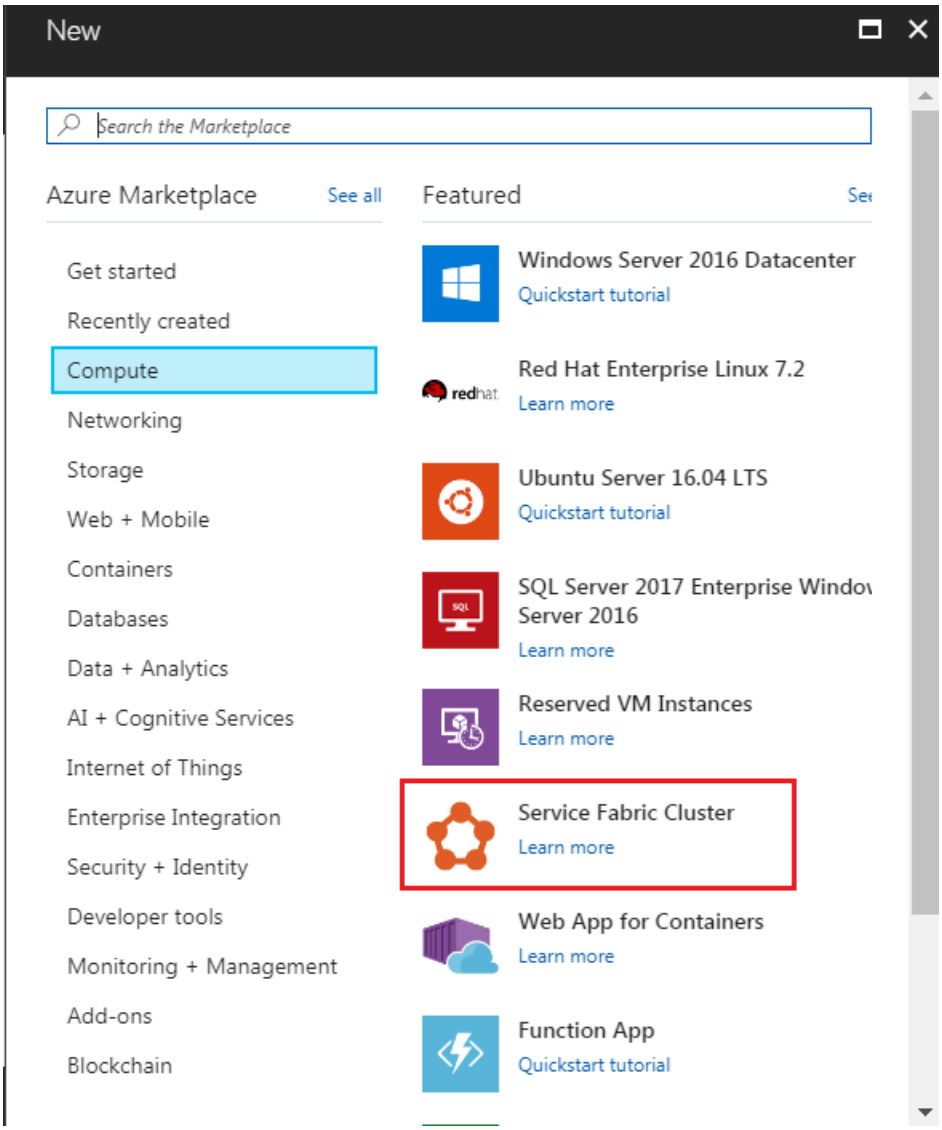
LESSONS LEARNED PROS/CONS

- Customizable and fast deployment can alleviate the problems of standing up VMs for every need
- In my experience, can be a bit tricky to implement properly

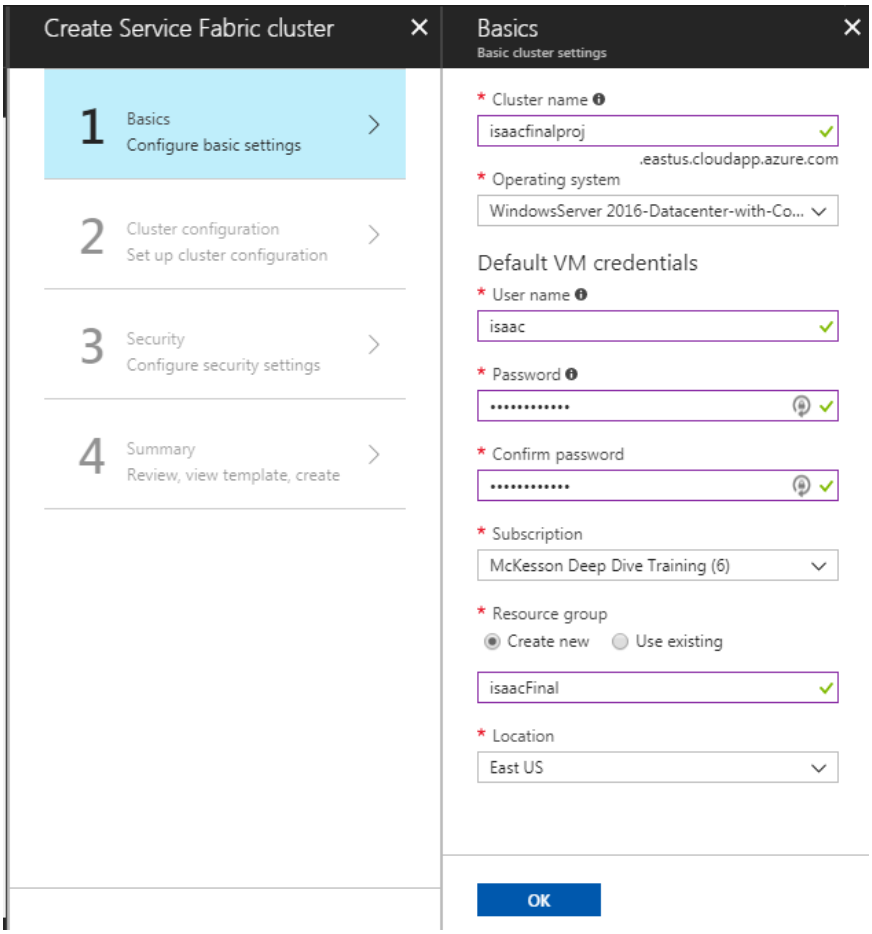
Github link: https://github.com/mck-isaac/azurefinal

Install the Service Fabric SDK from Microsoft. The SDK allows for the development of Service Fabric services and applications. https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-get-started

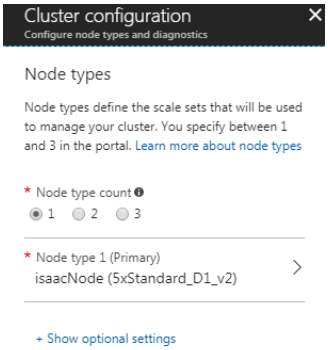
Next, create a Service Fabric cluster using Azure Portal. Click New > Compute > Service Fabric Cluster

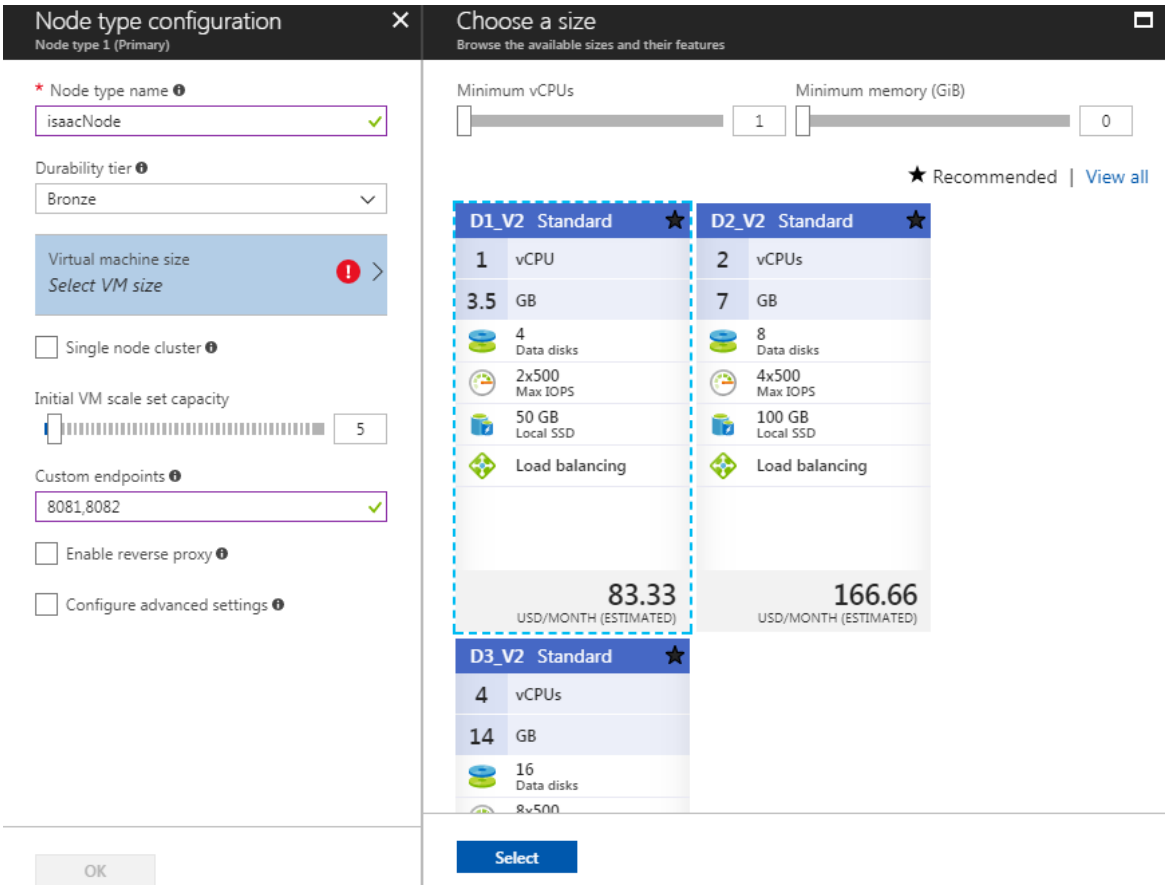


This will bring up the setup. In the Basics section, fill in the cluster information:

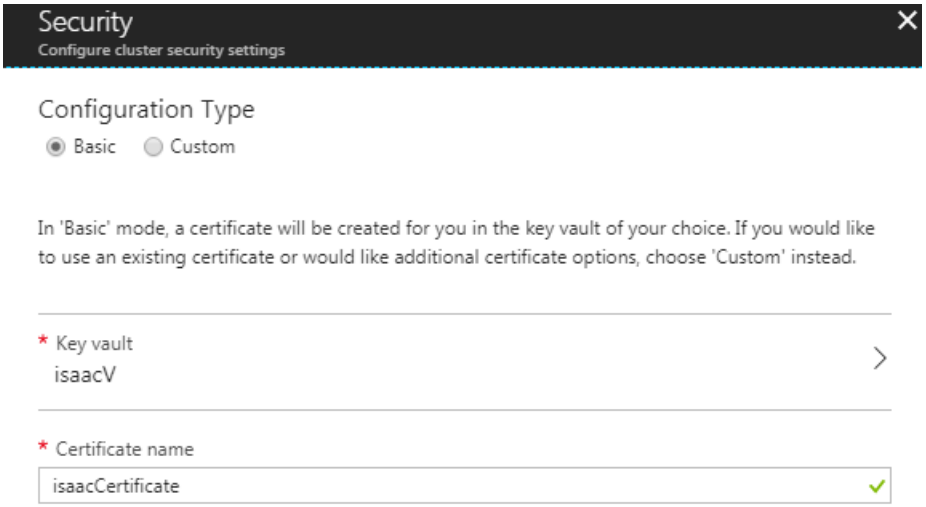


Next is the Cluster Configuration. There can be multiple different node type setups if there is a need to have different types of VMs within the cluster.

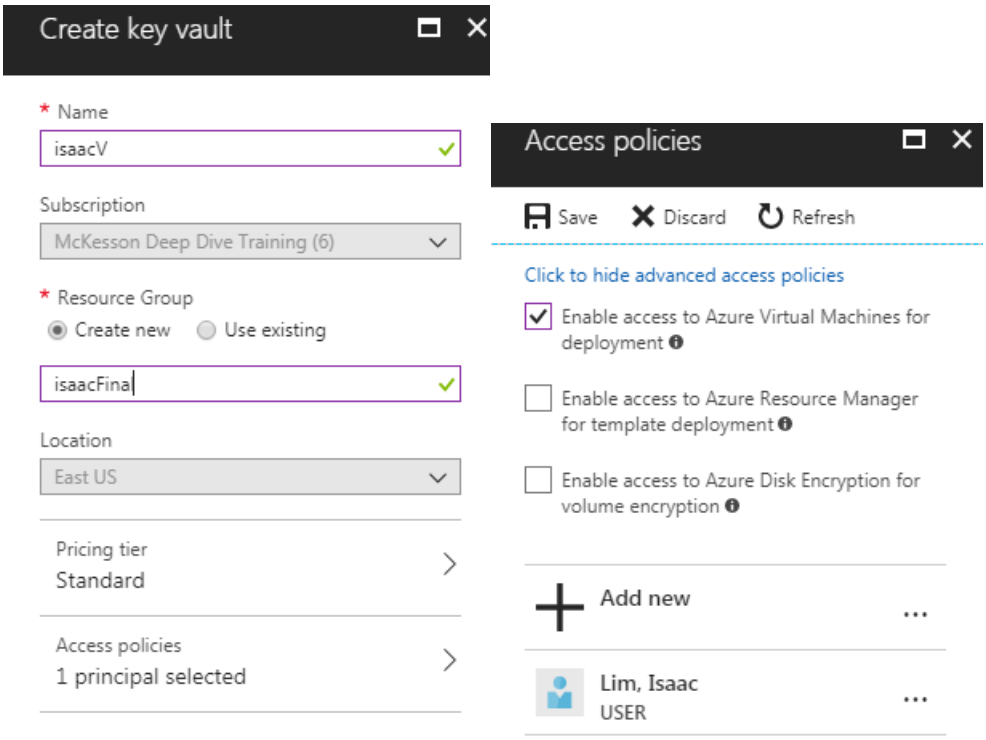




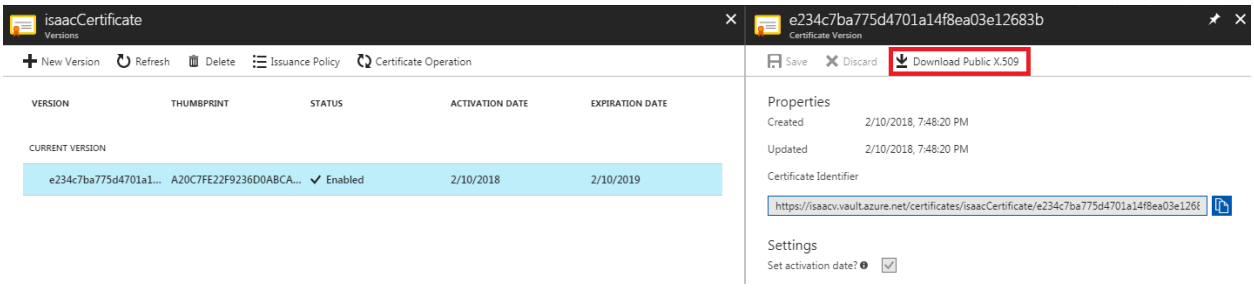
In the Security settings, a key vault needs to be used in order to generate and store the certificates that the cluster will be using:



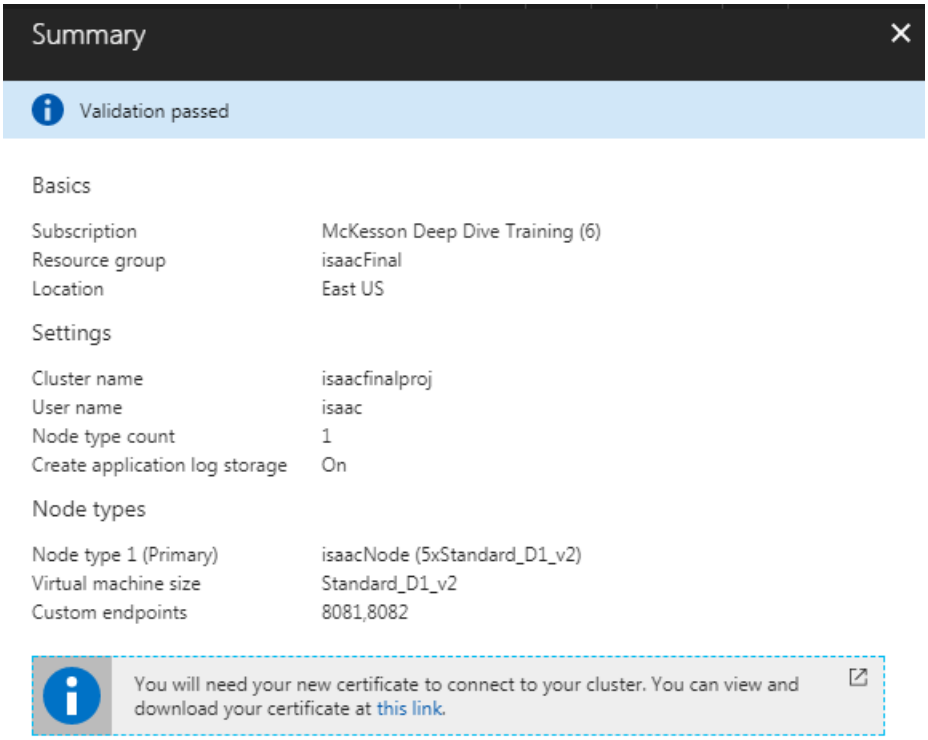
When creating the key vault, be sure the box for Enable access to Azure Virtual Machines for deployment is checked.



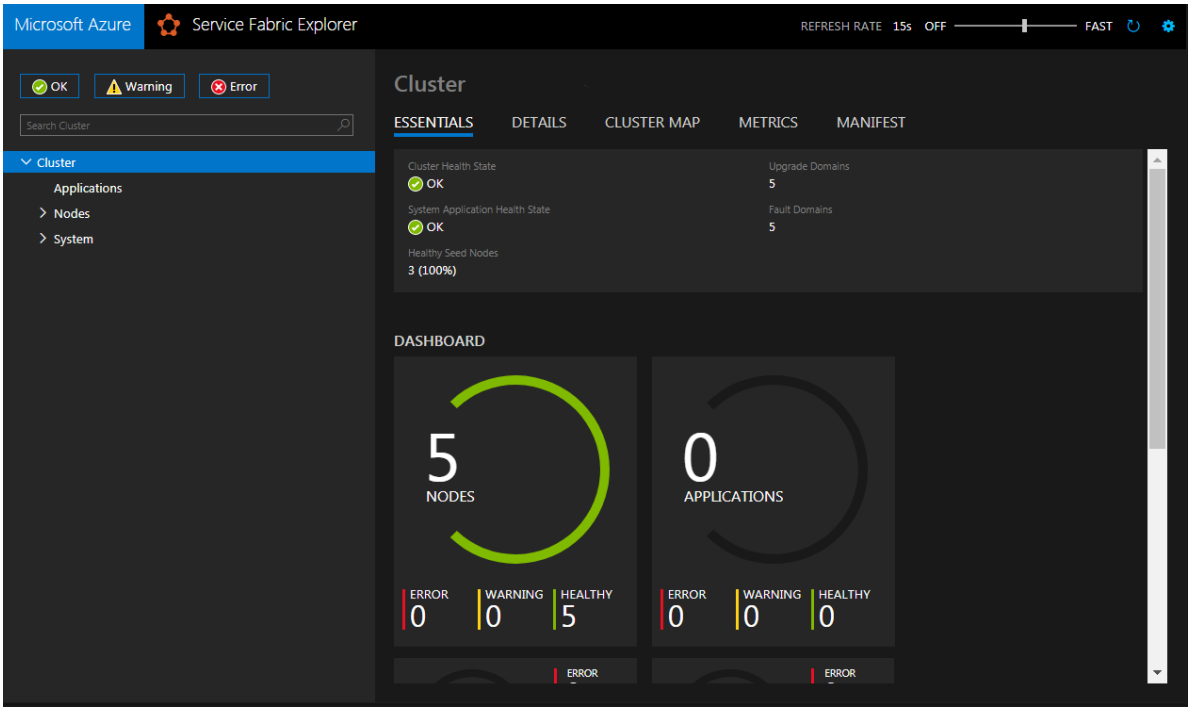
Download the certificate from the vault and be sure to install it on the client machine:



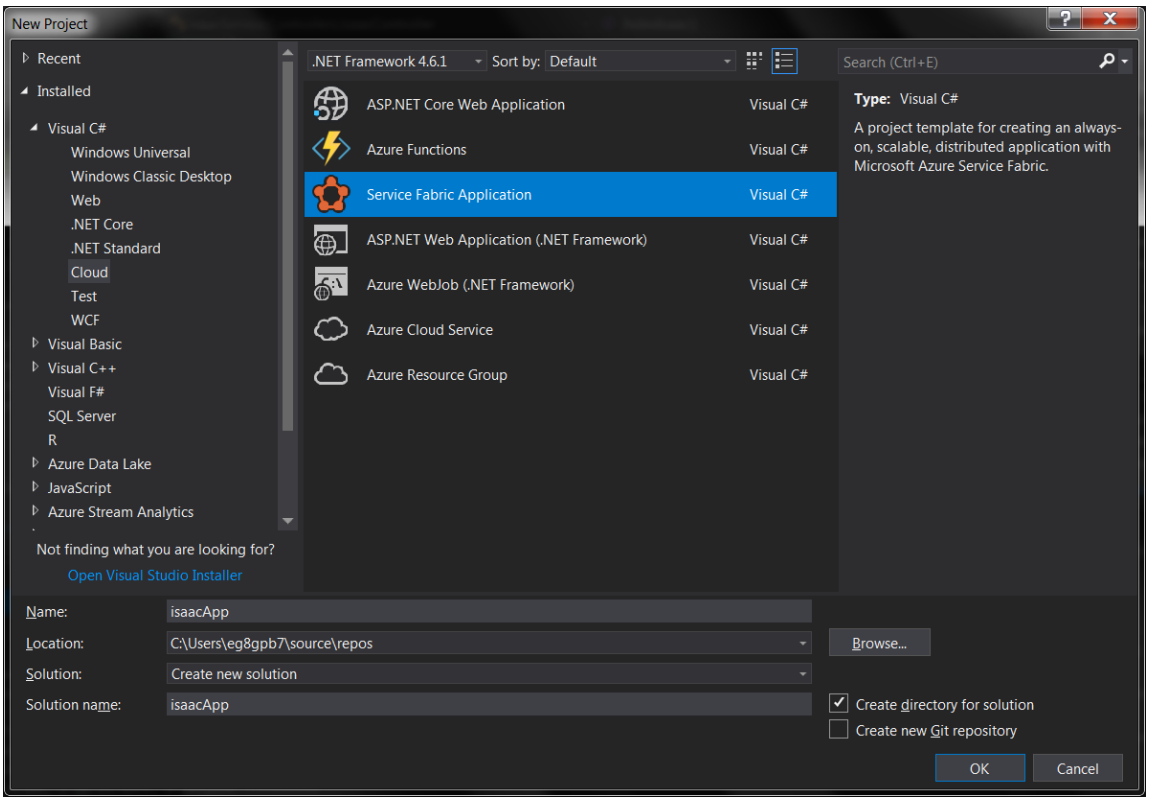
This deployment takes some time as it is creating 5 VMs to make up the cluster.



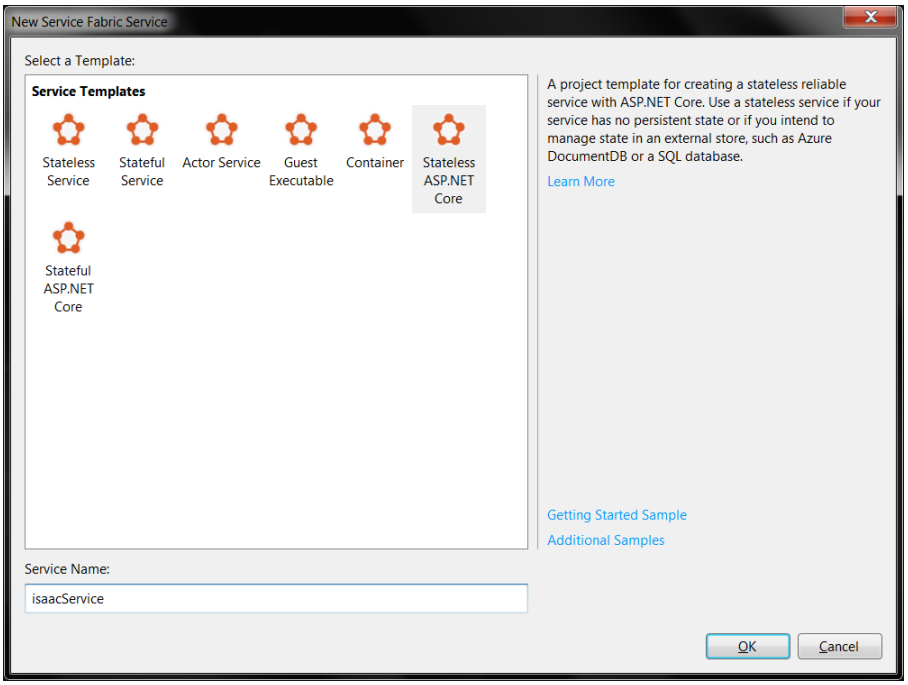
Once the cluster is deployed, the explorer can be accessed from the browser using port 19080.
https://isaacfinalproj.eastus.cloudapp.azure.com:19080/Explorer



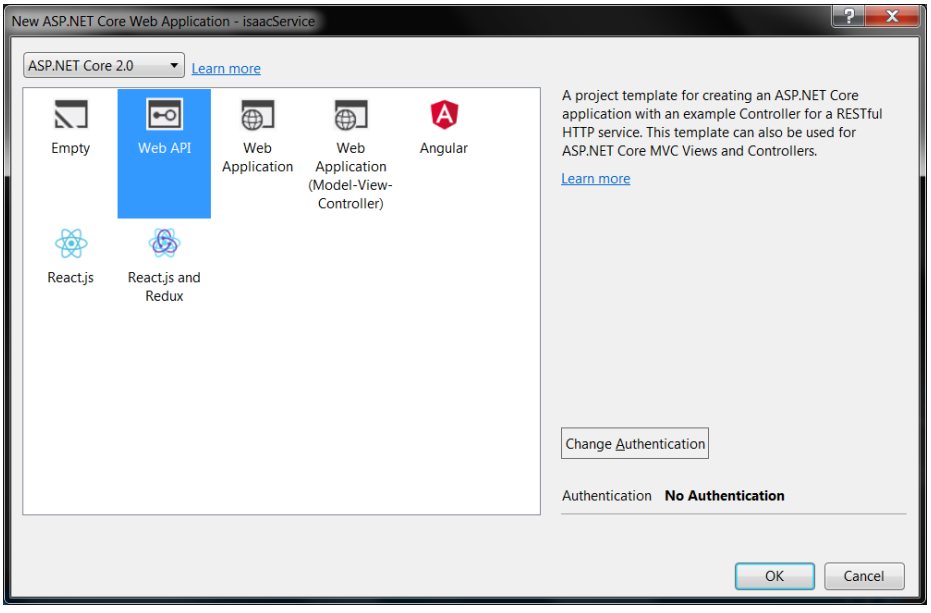
To create the application, go to Visual Studio. File > New > Project:



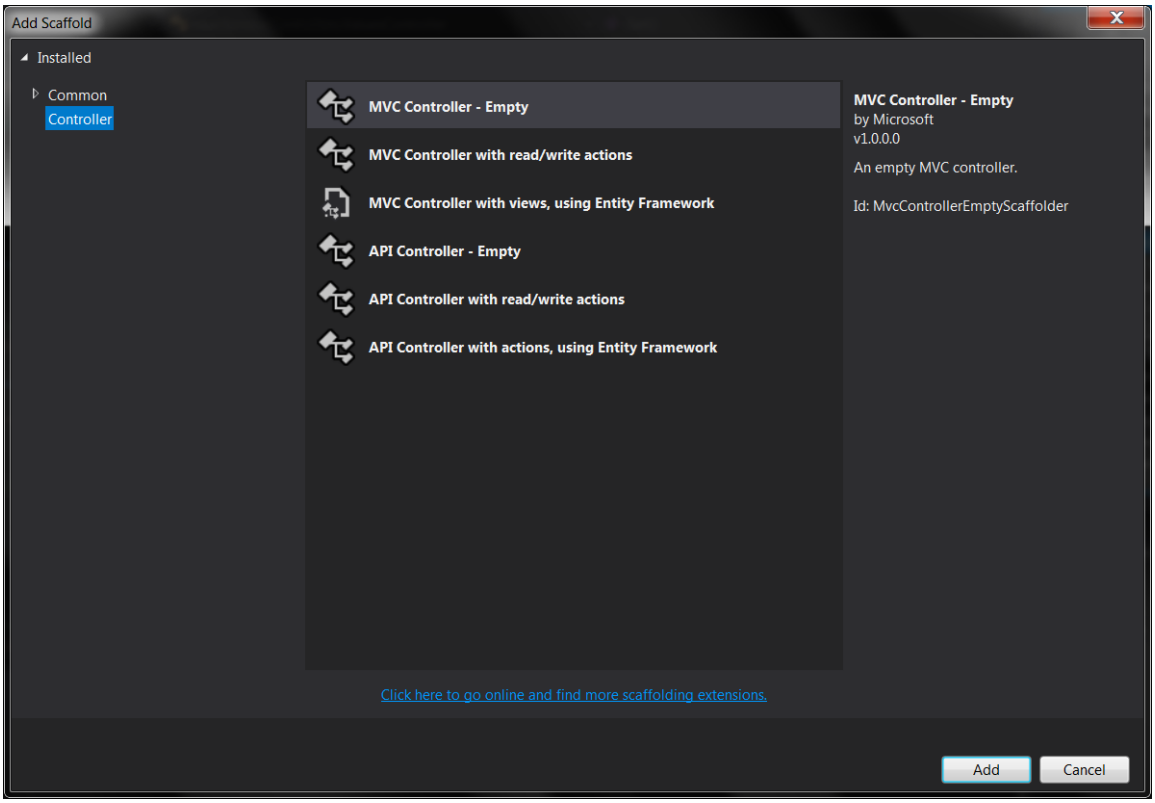
Next it will prompt to create the Service Fabric service, choose Stateless ASP.NET Core:



Also choose what kind of Web App, Web API:



This takes some time to create the project. Create a new Controller to access the data. Expand the service application (isaacService) > right click on Controllers folder > Add > Controller. Create an MVC Controller:

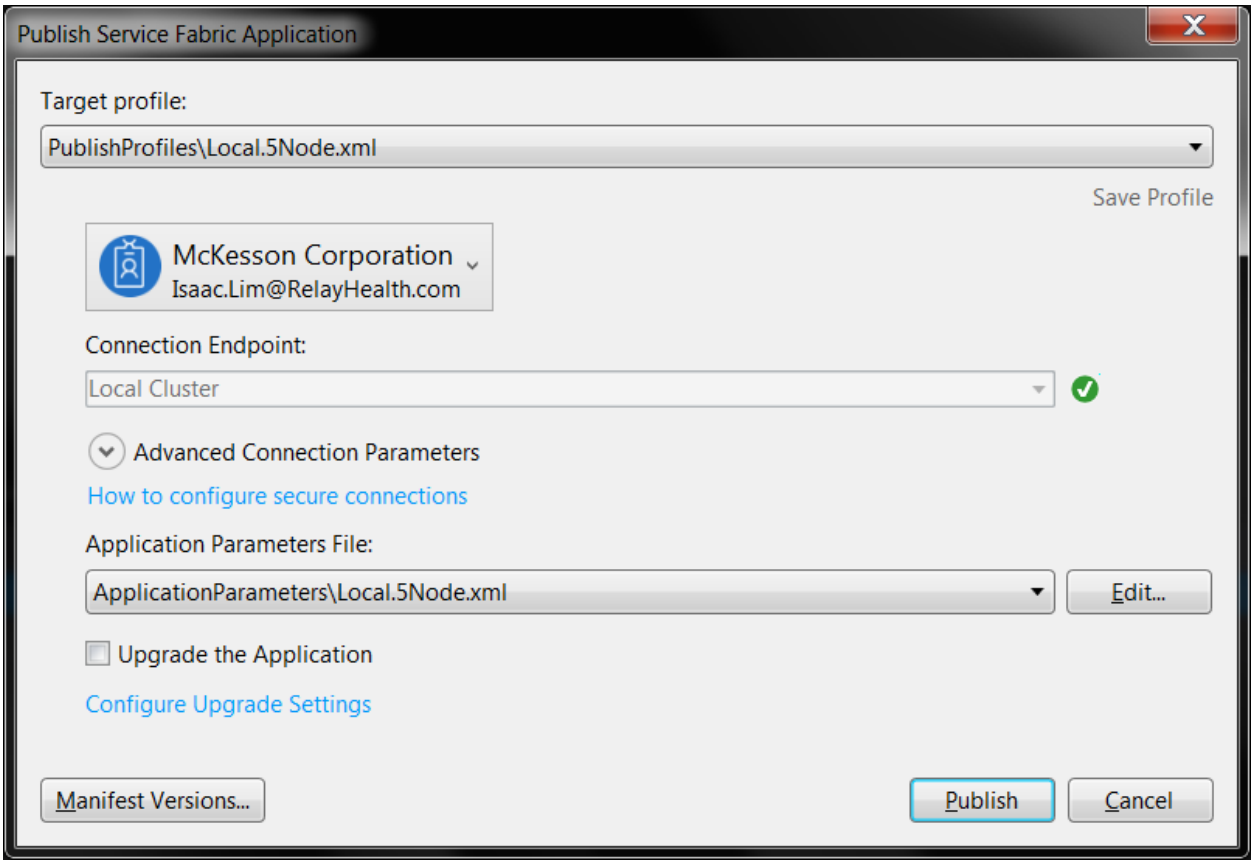


The relevant code is below that will read the dataset and return a random record as a JSON object.

```
namespace isaacService.Controllers
{

    class Cause
    {
        public String Year { get; set; }
        public String Name { get; set; }
        public String State { get; set; }
        public String Deaths { get; set; }
        public String Age { get; set; }
    }
   [Route("api/[Controller]")]
    public class isaacController : Controller
    {

        List<Cause> causes = new List<Cause>();

        public IActionResult IndexIsaac()
        {
            var fileName = @"NCHS_-_Leading_Causes_of_Death__United_States.csv";
            var file = System.IO.File.ReadLines(fileName).ToList();
            int count = file.Count();
            Random rnd = new Random();
```

```
            int skip = rnd.Next(0, count);
            string line = file.Skip(skip).First();
            String[] contents = line.Split('|');

            Cause cause = new Cause()
            {
                Year = contents[0],
                Name = contents[2],
                State = contents[3],
                Deaths = contents[4],
                Age = contents[5]
            };

            causes.Add(cause);

            return new JsonResult(causes);

        }
```

Right click on the Solution and click Build Solution. When that builds successfully, right click on the app (isaacApp) and click Publish:



Here is a sample of the data used:

	Year	113 Cause Name	Cause Name	State	Deaths	Age-adjusted Death Rate
2	1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Alabama	2313	52.2
3	1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Alaska	294	55.9
4	1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Arizona	2214	44.8
5	1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Arkansas	1287	47.6
6	1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	California	9198	28.7
7	1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Colorado	1519	39
8	1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Connecticut	1034	29.3
9	1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Delaware	267	35.3
10	1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	District of Columbia	161	28.4
11	1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Florida	5961	35.7
12	1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Georgia	3078	41.5
13	1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Hawaii	293	24.3
14	1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Idaho	597	48.3
15	1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Illinois	4125	33.7
16	1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Indiana	2309	38.4
17	1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Iowa	1123	35.2
18	1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Kansas	1126	40.7
19	1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Kentucky	1730	43.3
20	1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Louisiana	1940	44.7

The display of on the web service would have been the contents of one of those rows picked at random.

If everything is setup correctly, the service is deployed across the 5 nodes, and the controller is now accessible. Unfortunately for my case, I was not able to get this configured 100% correctly. I had multiple issues, mainly dealing with the site certificates. I could not get it to work, and that lead the result not being able to be displayed.