Building Apps for the Intelligent Cloud Day 2 – Service Fabric DEMOS

In this document, you will find guidance on what to demo to Partners, as well a reference video containing overview for service fabric features covered in each demo, and finally the step by step showing how to set up each demo covered in the slides.

For the sake of the actual application code to be used for service fabric applications, you can find a great deal of super cool applications that can be especially showcased for service fabric over [here - Service Fabric Samples](https://azure.microsoft.com/en-in/documentation/samples/service-fabric-dotnet-getting-started/).

At the end of this document, you will find the reference videos that you can follow to get a better visual sense of how these demos can be demonstrated. Most of the demos here, are based on these video references.

# Building your first microservice with service fabric

Requirement: Visual Studio 2013 or higher, Azure SDK

Description: Demo the creation of a simple service fabric application from visual studio and familiarize the audience with the developer tools, including the visual studio templates and service fabric cluster visualizer

Demo Time Estimation: 10 Minutes

What to demo:

1. Start by opening visual studio and going over the default service fabric templates available under the cloud section of File🡪NewProject
2. Choose any one template (like stateful service) and explore the project structure
3. Start to debug the app by pressing F5
4. Talk about local cluster and why deploying takes a while the first time, as Visual Studio is creating a local cluster for development. A local cluster runs the same platform code that you will build on in a multi-machine cluster, just on a single machine. The cluster creation status displays in the Visual Studio output window.
5. Take the screen to the service fabric cluster explorer and talk about it. [More Info](https://azure.microsoft.com/en-in/documentation/articles/service-fabric-visualizing-your-cluster/)
6. Talk about the rich diagnostics that become available as soon as we start to debug the application. Open the diagnostics event viewer (VS automatically brings it up) and expand one particular event to find more details.
7. End by talking about the impact of cleaning the local cluster and recommend to do that only if the developer testing is done, as it will take time for the next local run to create the cluster.

Reference Source and video:

* <https://azure.microsoft.com/en-in/documentation/articles/service-fabric-create-your-first-application-in-visual-studio/>

# create your service fabric cluster in azure

Requirement: Azure Subscription

Description: Demo how to create a service fabric cluster in Azure

Demo Time Estimation: 10 Minutes

What to demo:

* Show how easy it is to create service fabric cluster from the Azure Portal.
* For the simplicity of the demo, you can choose to go with the unsecure cluster as well
* Search for service fabric cluster in Azure portal, and create one by walking through the steps.

Reference Source:

* <https://azure.microsoft.com/en-us/documentation/articles/service-fabric-cluster-creation-via-portal/>

# creating a reliable service application

Requirement: Visual Studio 2013 or higher, and Azure Subscriptions

Description: Demo how you can create reliable services with in-built high availability using the Reliable Services programming model

For the purpose of this demo, you can use the Chatter application or the WordCount application that you can find from [here](https://azure.microsoft.com/en-in/documentation/samples/service-fabric-dotnet-getting-started/).

Demo Time Estimation: 15 Minutes

What to Demo:

* Open the Chatter application in visual studio
* Walkthrough the code that contains the reliable service APIs
* Explain how reliable collections are used in stateful services, and talk about its use for storing hot data
* Run the application locally and show the chat application
* Disable one node and show how the primary automatically fails over to a secondary from the cluster explorer
* After doing this, come back to the chatter application in the browser and show that we have not incurred any data loss, and that all the recent chats are preserved.

Reference Video/Demo Set up:

* Video: Watch after 54th minute of <https://mva.microsoft.com/en-US/training-courses/build-always-on-hyper-scalable-microservice-based-cloud-services-13992>
* Overview: <https://azure.microsoft.com/en-us/documentation/articles/service-fabric-reliable-services-quick-start/>

# upgrading a service

Requirement: Visual Studio 2013 or higher, VisualObjects demo application

Description: Showcase how seamless upgrades are made possible through service fabric, with zero downtime on the service/application

For the purpose of this demo, please download the visual objects sample application from [here](https://azure.microsoft.com/en-in/documentation/samples/service-fabric-dotnet-getting-started/).

And follow the detailed steps as given in the article [here](https://azure.microsoft.com/en-in/documentation/articles/service-fabric-application-upgrade-tutorial/#next-steps).

Demo Time Estimation: 10 Minutes

What to Demo:

* Deploy the visual objects solution from visual studio to the local cluster
* Open the visual objects application in internet explorer and dock it in one side
* Change the code in the VisualObjectActor.cs file as mentioned in the article above
* Make the changes in the versioning of the application
* Redeploy the application, but this time, make sure to check the Upgrade check box
* Show how the upgrade progresses from the cluster explorer
* More importantly, show how the application changes dynamically in internet explorer without any downtime

Reference Video/Demo Set up:

* Video: Watch after 45th minute of <https://mva.microsoft.com/en-US/training-courses/build-always-on-hyper-scalable-microservice-based-cloud-services-13992>
* <https://azure.microsoft.com/en-in/documentation/articles/service-fabric-application-upgrade-tutorial/#next-steps>