

Container App & Azure Service Fabric

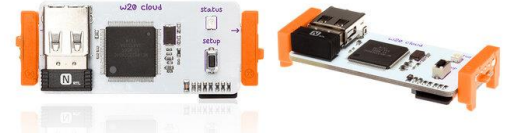
China CAT, Microsoft

Agenda

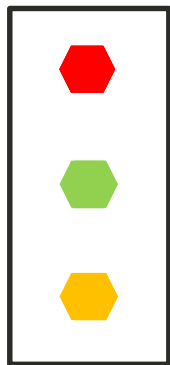
- Microservice in Service Fabric
- Container and Docker
- Demo: Container App in Service Fabric
- Q&A

Microservices?

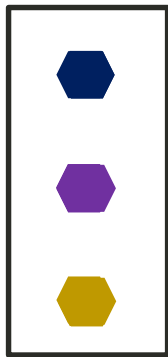
- Small and lightweight, easy to replace
- Compatibility with languages, databases and technologies
- Scalability, Partitioning, Versioning, CI/CD
- Communication to fulfill a goal
- Concrete and modern interpretation of [service-oriented architectures](#) (SOA)



Microservices in Service Fabric

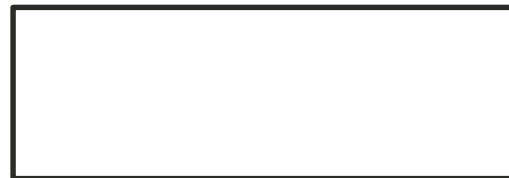
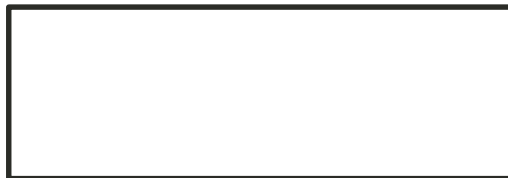


App1



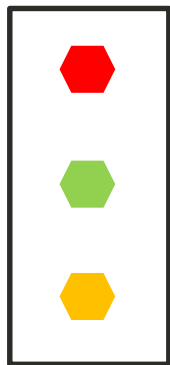
App2

App Type Packages

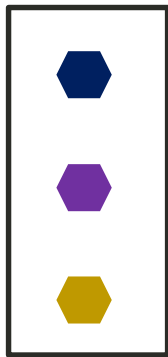


Service Fabric Cluster VMs

Handling Machine/Node Failures

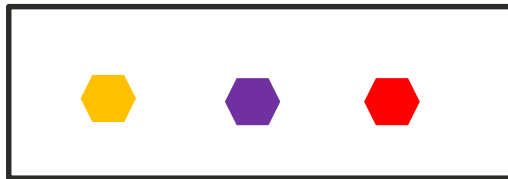
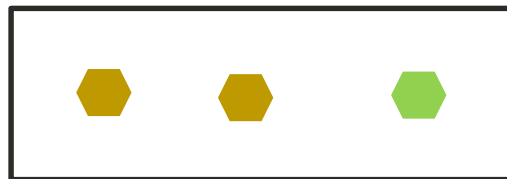
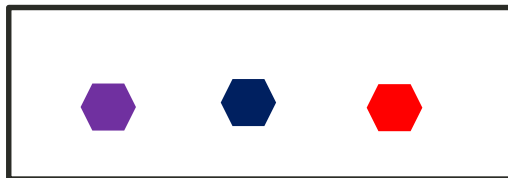


App1



App2

App Type Packages



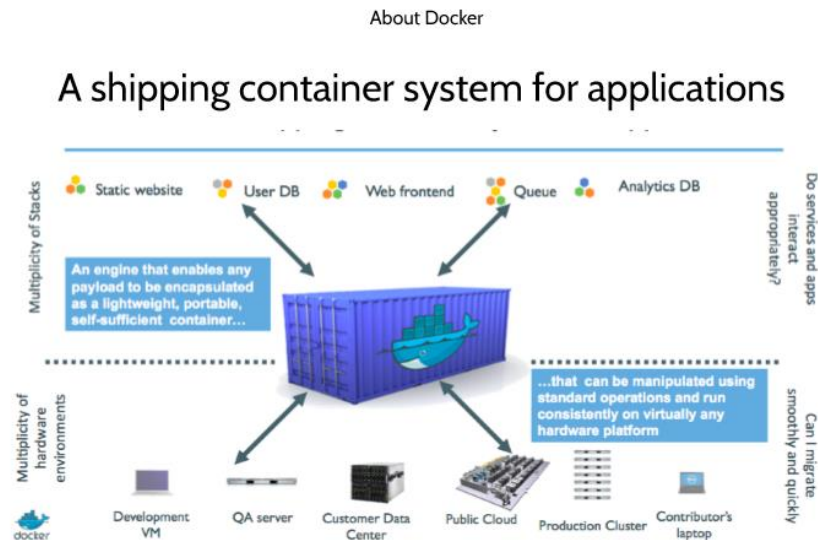
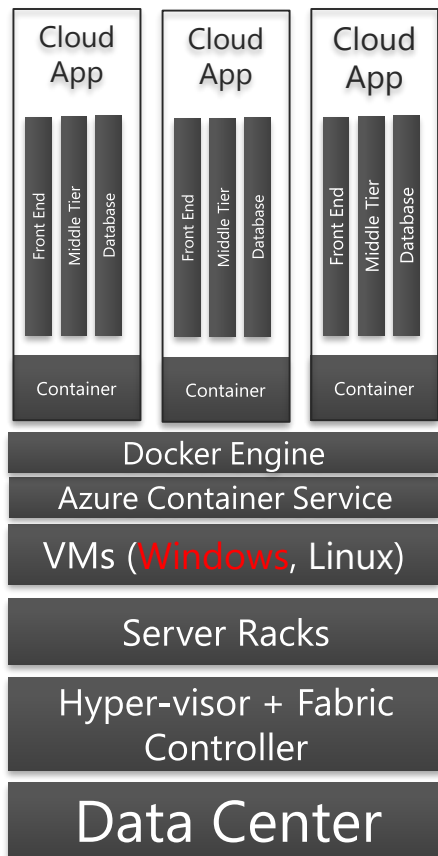
Service Fabric Cluster VMs

Why Service Fabric

- Fast time to market (fine-grained service, more agility)
- Host always-on, scalable, reliable, data-intensive, distributed applications (partitions/replicas, reliable services)
- Deliver low-latency performance and efficiency at massive scale (reliable collection)
- Programming models for active “things” (actor)
- Isolate application with infrastructure, runnable in any cloud or on premise
- Container image as Microservice

What is Docker?

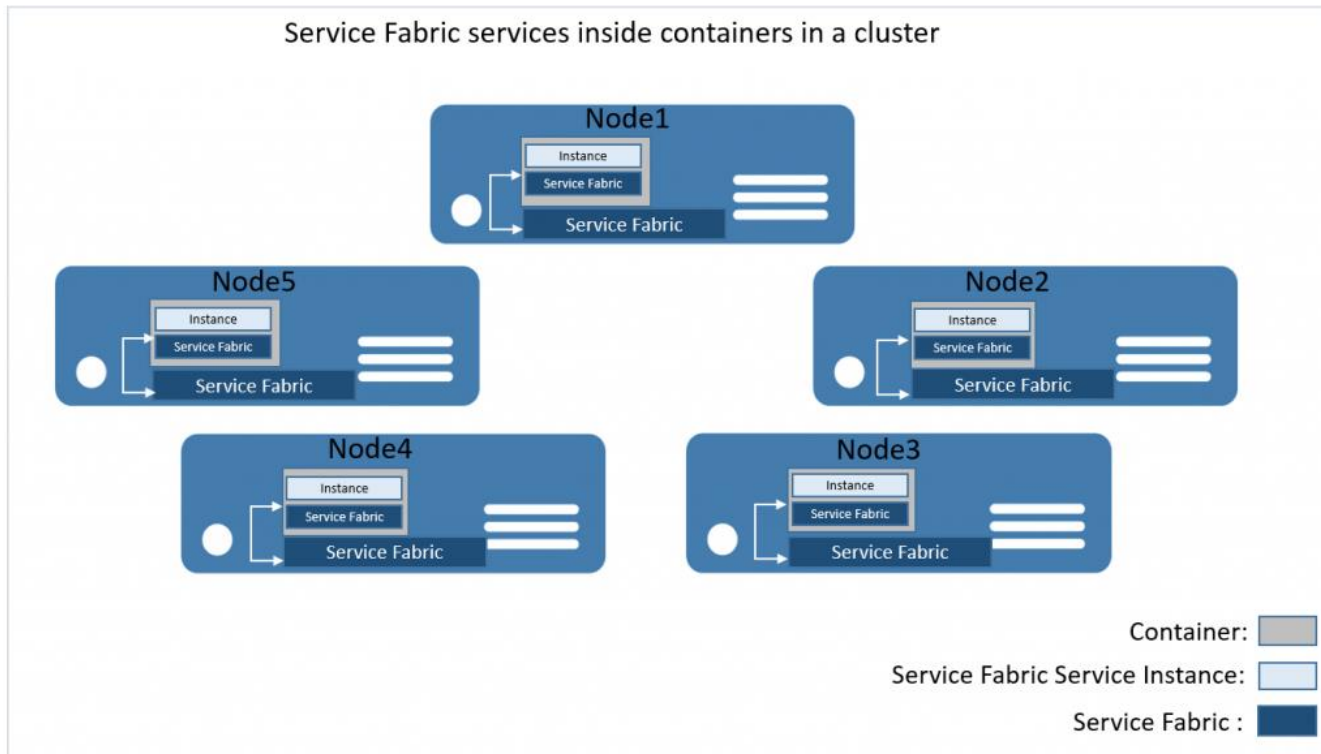
- An Engine that allows applications to be portable and deployed in self-sufficient containers.
- Packs a set of containers on a host
- Applications can be built iteratively in an agile manner



Let's ask :

- Container App on Service Fabric?
- Benefits from running container app in SF?
- Efforts to CI/CD?

Container App in Service Fabric



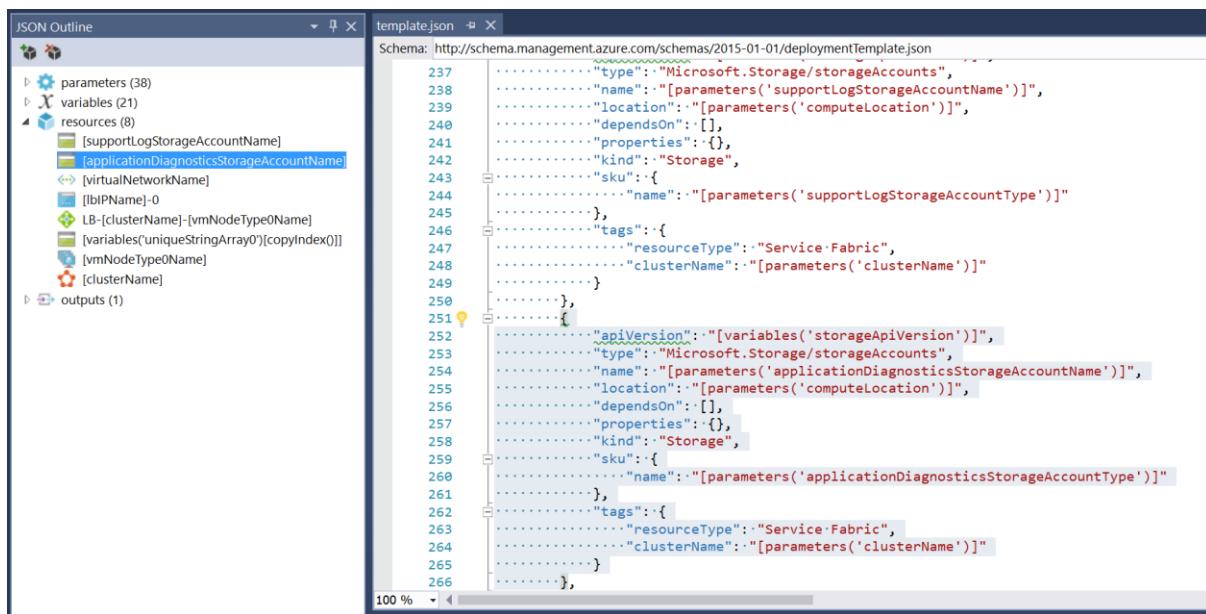
Demo: Container App in Service Fabric

1. Define customer cluster and generate template (Azure portal)
2. Prepare CI/CD devbox in Ubuntu server and create service fabric cluster from template
3. Connect service fabric cluster and make incremental upgrade
4. Prepare Azure Container Registry and push local Docker image into the registry
5. Create service fabric guest container app with specified Docker image
6. Deploy and manage container app into service fabric cluster by pure commands
7. Verify the result of running Docker image/app in service fabric cluster
8. Conduct cluster upgrade by commands and ensure no impact to running app

Define customer cluster and generate template (Azure portal)

Step 1:

<http://portal.azure.com/>



Step 2:

Prepare CI/CD devbox in Ubuntu server and create service fabric cluster from template

```
#Linux SF Cluster with CLI
```

```
az login
```

```
az account list
```

```
az account set --subscription 73cafea5-b590-40da-b211-3e9fb8dd3b6d
```

```
azure config list --json
```

```
azure config mode 'arm'
```

```
azure group create --name tdcdemo001-rg --location eastasia
```

```
azure group list
```

```
azure group deployment create --name "sflinuxdep01" --resource-group "tdcdemo001-rg" --template-file  
"./template.json" --parameters-file "./parameters.json" --verbose
```

```
azure group deployment list --resource-group tdcdemo001-rg
```

Connect service fabric cluster and make incremental upgrade

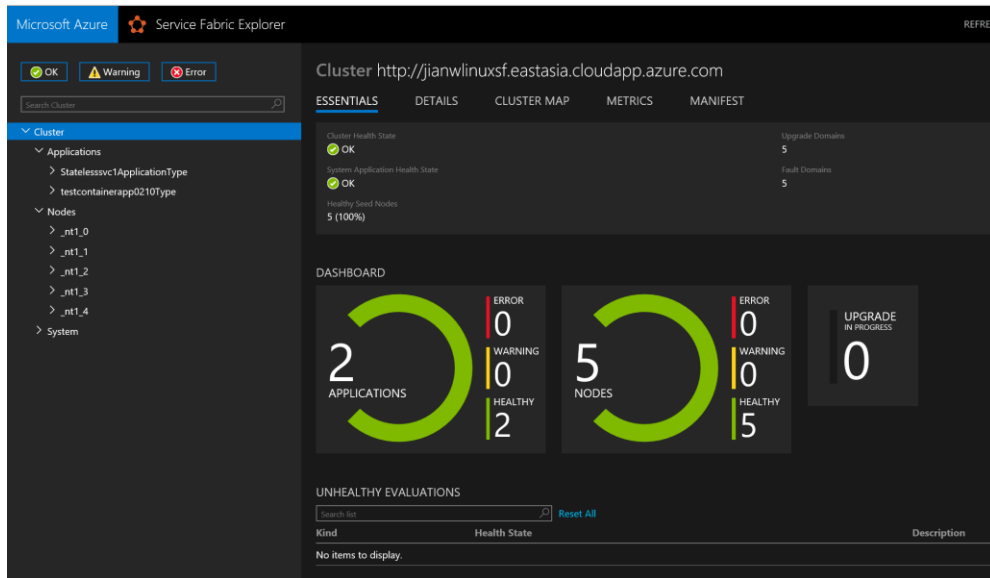
Step 3:

```
azure group deployment list --resource-group tdcdemo001-rg
```

```
azure group deployment create --mode Incremental --name "sflinuxdep01" --resource-group "tdcdemo001-rg" --  
template-file "./template2.json" --parameters-file "./parameters.json" --verbose
```

```
# delete group
```

```
#azure group delete --name tdcdemo001-rg
```



Step 4:

Prepare Azure Container Registry and push local Docker image into the registry

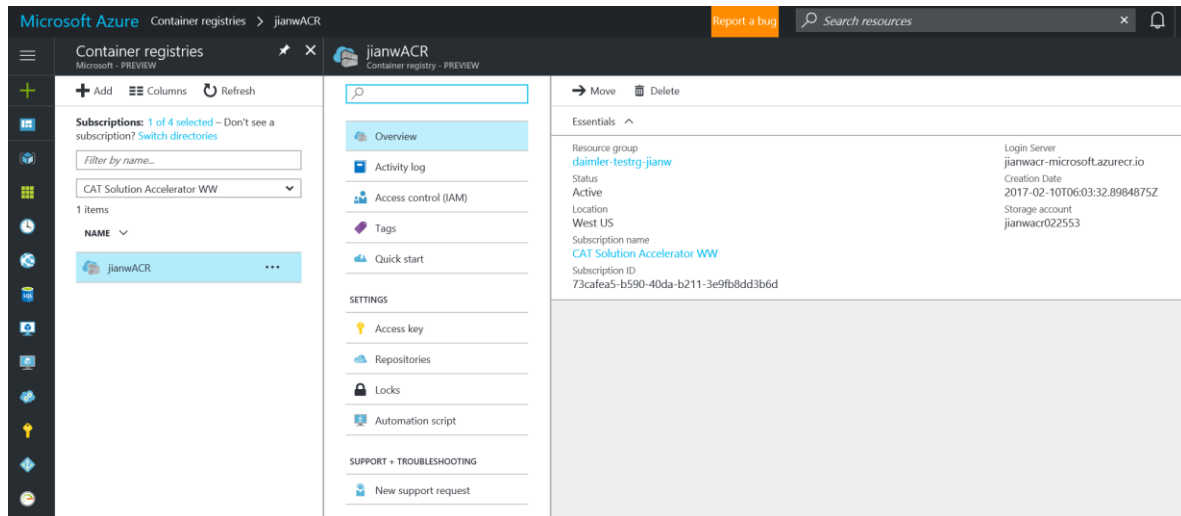
```
docker -H localhost:2375 images
#download nginx image
sudo docker pull nginx
```

```
sudo docker login jianwacr-microsoft.azurecr.io -u jianwacr -p R+d/p/jm/G//uGX5Kqi/z7+HdBZl10Mq
```

```
docker -H localhost:2375 images
sudo docker tag nginx jianwacr-microsoft.azurecr.io/demos/nginx001
sudo docker push jianwacr-microsoft.azurecr.io/demos/nginx001
```

Manage ACR Images

<http://aka.ms/acr/manage> → <https://acrwebmanagerdemo.azurewebsites.net/>



The screenshot shows the Microsoft Azure portal interface for managing Container Registries. The main heading is "Container registries" with a sub-heading "jianwacr". The left sidebar shows a list of registries with one item, "jianwacr". The right-hand pane displays the "Overview" tab for the "jianwacr" registry, showing details such as Resource group (daimler-testrg-jianw), Status (Active), Location (West US), Subscription name (CAT Solution Accelerator WW), and Subscription ID (73cafea5-b590-40da-b211-3e9fb8dd3b6d). The left-hand pane also shows a search bar and a list of registries.

Create service fabric guest container app with specified Docker image

Step 5:

```
yo azuresfguest
```

```
travisye/backend:latest  
https://hub.docker.com/r/travisye/backend/
```

```
docker0707/dockersf  
https://hub.docker.com/r/docker0707/dockersf/
```

```
jianwacr-microsoft.azurecr.io/demos/nginx001
```

```
testadmin@ubuntu-sfdevcluster:~$ az acr repository list --name jianwacr  
Result  
-----  
demos/nginx001  
demos/nginx002  
testadmin@ubuntu-sfdevcluster:~$
```

```
azureuser@ubuntu-sfdevcluster:~$ sudo docker tag 573caec6a40d cattest-microsoft.azurecr.io/samples/testimage  
azureuser@ubuntu-sfdevcluster:~$ sudo docker images  
REPOSITORY                                TAG      IMAGE ID      CREATED      SIZE  
cattest-microsoft.azurecr.io/samples/testimage  latest  573caec6a40d  8 days ago  650.3 MB  
docker0707/dockersf                        latest  573caec6a40d  8 days ago  650.3 MB  
azureuser@ubuntu-sfdevcluster:~$
```

Deploy and manage container app into service fabric cluster by pure commands

Step 6:

#deploy app

azure servicefabric cluster connect --connection-endpoint <http://sfdemo001.eastasia.cloudapp.azure.com:19080>

```
13.75.126.187
testadmin@ubuntu-sfdevcluster:~/sf-practices-yo/tdcdemo/testcontainerapp0210$ find ./
./
./install.sh
./testcontainerapp0210
./testcontainerapp0210/myservice1Pkg
./testcontainerapp0210/myservice1Pkg/code
./testcontainerapp0210/myservice1Pkg/code/Dummy.txt
./testcontainerapp0210/myservice1Pkg/config
./testcontainerapp0210/myservice1Pkg/config/Settings.xml
./testcontainerapp0210/myservice1Pkg/ServiceManifest.xml
./testcontainerapp0210/ApplicationManifest.xml
./uninstall.sh
```


Verify the result of running Docker image/app in service fabric cluster

Step 7:

```
docker -H localhost:2375 images  
docker -H localhost:2375 ps
```



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Conduct cluster upgrade by commands and ensure no impact to running app

Step 8:

```
azure group deployment create --mode Incremental --name "sflinuxdep01" --resource-group "tdcdemo001-rg" --template-file "./template3.json" --parameters-file "./parameters.json" --verbose
```

```
13.75.126.187
testadmin@ubuntu-sfdevcluster:~/sf-practices-yo/tdcdemo/template$ diff template2.json template3.json
197c197
<         "defaultValue": 5,
---
>         "defaultValue": 6,
testadmin@ubuntu-sfdevcluster:~/sf-practices-yo/tdcdemo/template$ █
```

Q & A

Jonathan Tong (jtong@microsoft.com)
Eason Lai (eason.lai@microsoft.com)
Jian Wu (jianw@microsoft.com)