

# JOURNEY TO THE CLOUD

A Developer's Perspective

**DIONNE CONDOR-FARRELL**

Senior Developer (Java) – Transport for London  
Founding Member - UKBlackTech

Twitter: @dionnecf

MY JOURNEY TO THE CLOUD...

Frustration with limited resources





# MY JOURNEY TO THE CLOUD...

## Disparity between development teams



A human skeleton is depicted sitting at a desk, leaning forward with its right hand resting on its chin in a contemplative pose. The skeleton is positioned in front of a black computer monitor. The background is a solid, dark grey. The text "MY JOURNEY TO THE CLOUD..." is overlaid in the upper right quadrant of the image.

MY JOURNEY TO THE CLOUD...

Having to rely on another team

# MY JOURNEY TO THE CLOUD...

## Research Cloud Computing Providers



And many more...



# BENEFITS OF EMBRACING THE CLOUD

- Saves Time
- Simplicity
- Flexibility
- Integration
- Save Your Sanity!

# Microsoft Azure



Azure App Service



Web Apps



Mobile Apps



API Apps

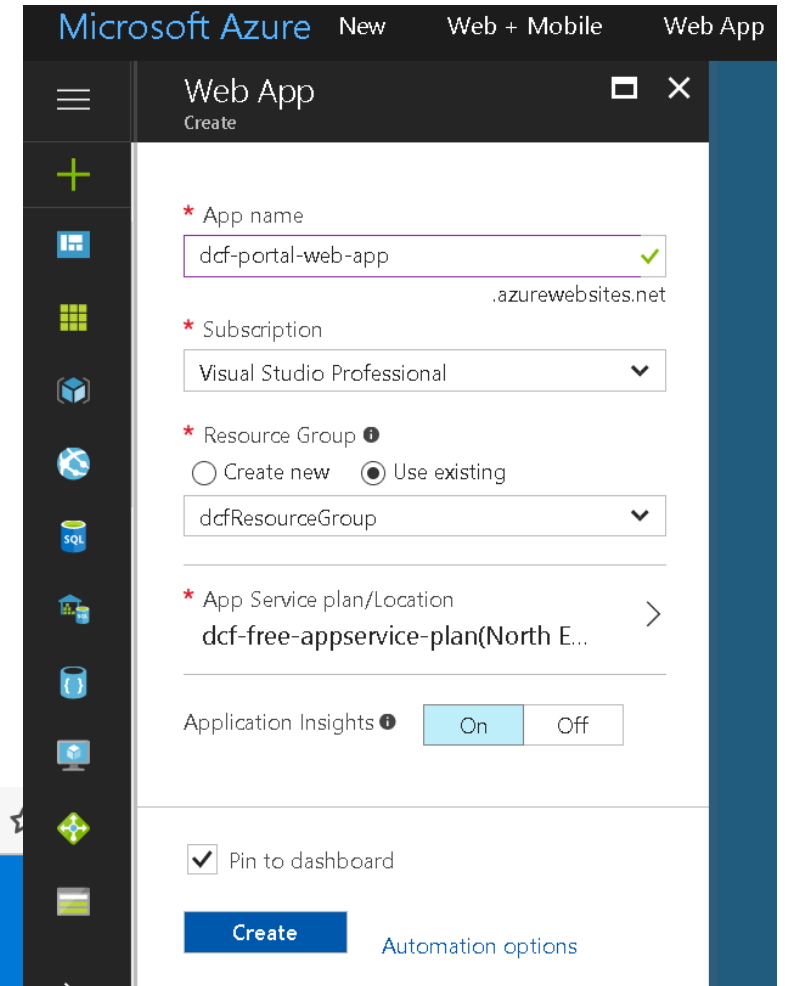
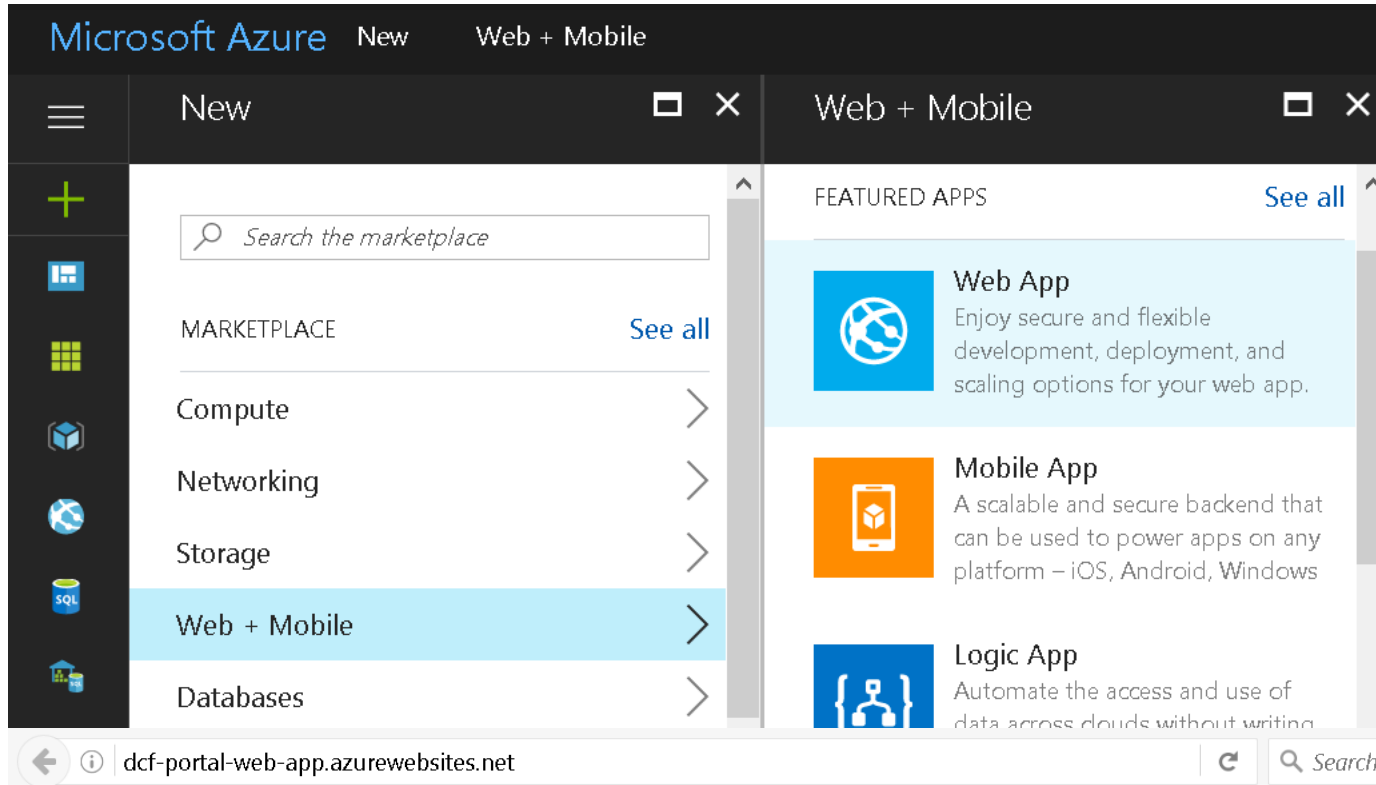


Logic Apps



# AZURE WEB APPS VIA PORTAL

<https://portal.azure.com>



Your App Service app has been created

Go to your app's **Quick Start** guide in the Azure portal to get started or read our [deployment documentation](#).

# AZURE WEB APPS VIA AZURE CLI

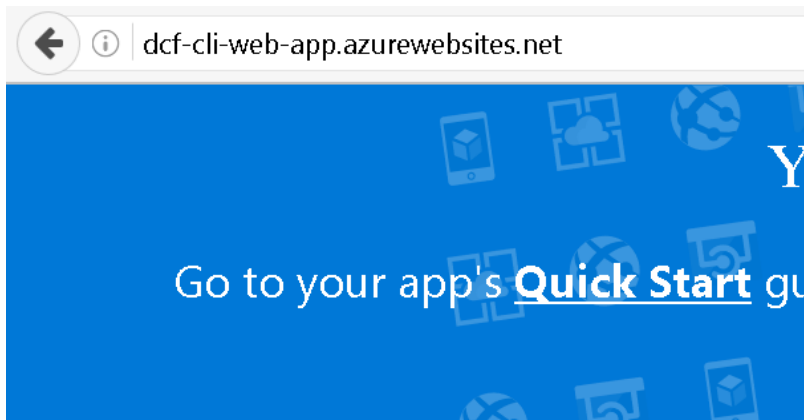
<https://docs.microsoft.com/en-us/cli/azure/install-azure-cli>

```
C:\Users\Dionne>az group create --location "North Europe" --name dcfResourceGroup
```

```
C:\Users\Dionne>az appservice plan create --name dcf-free-appservice-plan --resource-group dcfResourceGroup --sku FREE
```

```
C:\Users\Dionne>az appservice web create --name dcf-cli-web-app --resource-group dcfResourceGroup --plan dcf-free-appservice-plan
```

```
C:\Users\Dionne>az appservice web source-control config --name dcf-cli-web-app --resource-group dcfResourceGroup --repo-url "https://github.com/azure-appservice-samples/JavaCoffeeShopTemplate.git" --branch master --manual-integration
```



Cappuccino

\$3.45

Cappuccino

# AZURE WEBAPPS VIA TOOLKIT FOR IDE

The screenshot displays the IntelliJ IDEA interface with the 'Deploy to Azure Web App Container' dialog open. The 'Azure' menu is expanded, showing options like 'Configure Application Insights...', 'Publish as Azure Web App...', and 'Publish as Azure Cloud Service...'. The 'New Web App Container' dialog is also open, showing the 'Basic' tab with fields for DNS Label, Web Container, Subscription, Resource Group, and App Service Plan. The 'JDK' tab is also visible, showing options for deploying the default JDK or a 3rd party JDK.

**Deploy to Azure Web App Container**

Choose an existing Azure Web App container to deploy this application to:

- dcf-edipse-april (JRE 1.8.0\_73; TOMCAT 8.0)
- dcf-edipse-multi-web-apps (JRE 1.8.0\_73; TOMCAT 8.0)
- dcf-edipse-web-app (JRE 1.8.0\_73; TOMCAT 8.0)

**New Web App Container**

**Basic** | JDK

DNS Label: dcf-edipse-april30.azurewebsites.net

Web Container: Apache Tomcat 8 (Latest)

Subscription: Visual Studio Professional

Resource Group: dcfResourceGroup

App Service Plan: dcf-free-appservice-plan

[Pricing](#)

Location: North Europe

Pricing Tier: Free

Instance Size: Small

**JDK**

☒ Deploy the default JDK offered by Azure Web Apps service (JDK 8)

☐ Deploy a 3rd party JDK available on Azure:

Azul Zulu, OpenJDK 1.8.0\_u92

☐ Deploy my own JDK from this download location:

JDK zip archive URL:

Storage account key (if the URL above is a private blob):

(Note: For custom JDK, ideally you should have only one web app container per app service plan with small instance size.)



# AZURE WEBAPPS DEPLOYMENT

Azure Toolkit for IDEs



FTP or Open Source Version Control Repositories or Private Repositories

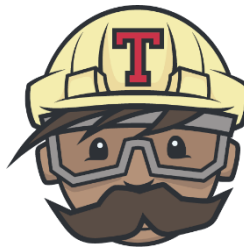


**GitHub**



**Bitbucket**

Integrate with your favourite Continuous Integration/Continuous Deployment tools



# AZURE VIRTUAL MACHINES



- 1 in 3 Azure VMs are running Linux  
e.g. Centos, Suse, Core OS VMs, Red Hat, Ubuntu
- Supports language like Node.JS, ASP.NET, Python, PHP, Java, Ruby
- Control over the Operating System, Software installation, Maintenance, Updates, Security, etc.
- Good for migrating existing applications to the Cloud Computing with “Lift and Shift” approach

# AZURE LINUX VM – BASICS AND SIZE

The screenshot displays the Microsoft Azure portal interface for creating a new virtual machine. The 'Compute' section is active, showing a list of operating systems. The 'Basics' tab is selected, and the 'Choose a size' panel is open.

**Operating System Options:**

- Windows Server 2016 Datacenter
- Red Hat Enterprise Linux 7.2
- Ubuntu Server 16.04 LTS (Selected)
- SQL Server 2016 SP1 Enterprise on Windows Server 2016
- Virtual machine scale set

**Basics Configuration:**

- Name: dionneTestVM
- VM disk type: HDD
- User name: dcfuser
- Authentication type: SSH public key (Selected), Password
- Password: [Masked]
- Confirm password: [Masked]
- Subscription: Visual Studio Professional
- Resource group: dcfVMResourceGroup
- Location: West Europe

**Choose a size Configuration:**

- Supported disk type: HDD
- Minimum memory (GiB): 0
- Minimum cores: 1
- Recommended sizes: D1\_V2 Standard, D1 Standard, A1 Standard

Size	Core	Memory (GB)	Data disks	Max IOPS	Local SSD	Load balancing
D1_V2 Standard	1	3.5	2	2x500	50 GB	Load balancing
D1 Standard	1	3.5	2	2x500	50 GB	Load balancing
A1 Standard	1	1.75	2	2x500	50 GB	Load balancing



# AZURE LINUX VM – VIA PORTAL

## SETTINGS AND SUMMARY

Microsoft Azure

New

Compute

Ubuntu Server 16.04 LTS

Create virtual machine

Settings

Create virtual machine

1 Basics Done

2 Size Done

3 Settings Configure optional features

4 Summary Ubuntu Server 16.04 LTS

Storage

Use managed disks

No Yes

\* Storage account (new) dcfmresourcegroup666

Network

\* Virtual network (new) dcfVMResourceGroup-vnet

\* Subnet default (10.0.0.0/24)

\* Public IP address (new) dionneTestVM-ip

\* Network security group (firewall) (new) dionneTestVM-nsg

Extensions

Extensions No extensions

High availability

\* Availability set None

Monitoring

Boot diagnostics Disabled Enabled

Guest OS diagnostics Disabled Enabled

\* Diagnostics storage account (new) dcfmresourcegroup616

OK

Microsoft Azure

New

Compute

Ubuntu Server 16.04 LTS

Create virtual machine

Summary

Create virtual machine

1 Basics Done

2 Size Done

3 Settings Done

4 Summary Ubuntu Server 16.04 LTS

Validation passed

Subscription Visual Studio Professional

Resource group dcfVMResourceGroup

Location West Europe

Settings

Computer name dionneTestVM

Disk type HDD

User name dcfuser

Size Standard D1 v2

Storage account (new) dcfmresourcegroup643

Managed No

Virtual network (new) dcfVMResourceGroup-vnet

Subnet (new) default (10.0.1.0/24)

Public IP address (new) dionneTestVM-ip

Network security group (firewall) (new) dionneTestVM-nsg

Availability set None

Guest OS diagnostics Disabled

Boot diagnostics Enabled

Diagnostics storage account dionneTestVM-diag743

OK Download template and parameters

# AZURE LINUX VM - CONNECT VIA PUTTY (SSH)

```
C:\Users\Dionne>az vm list --output table
Name                ResourceGroup      Location
-----
dionneTestVM        DCFVMRESOURCEGROUP westeurope
DionnesVM           DIONNECFDEMO       westeurope
```

```
dcfuser@dionneTestVM: ~
login as: dcfuser
dcfuser@5[REDACTED]3's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-71-generic x86_64)

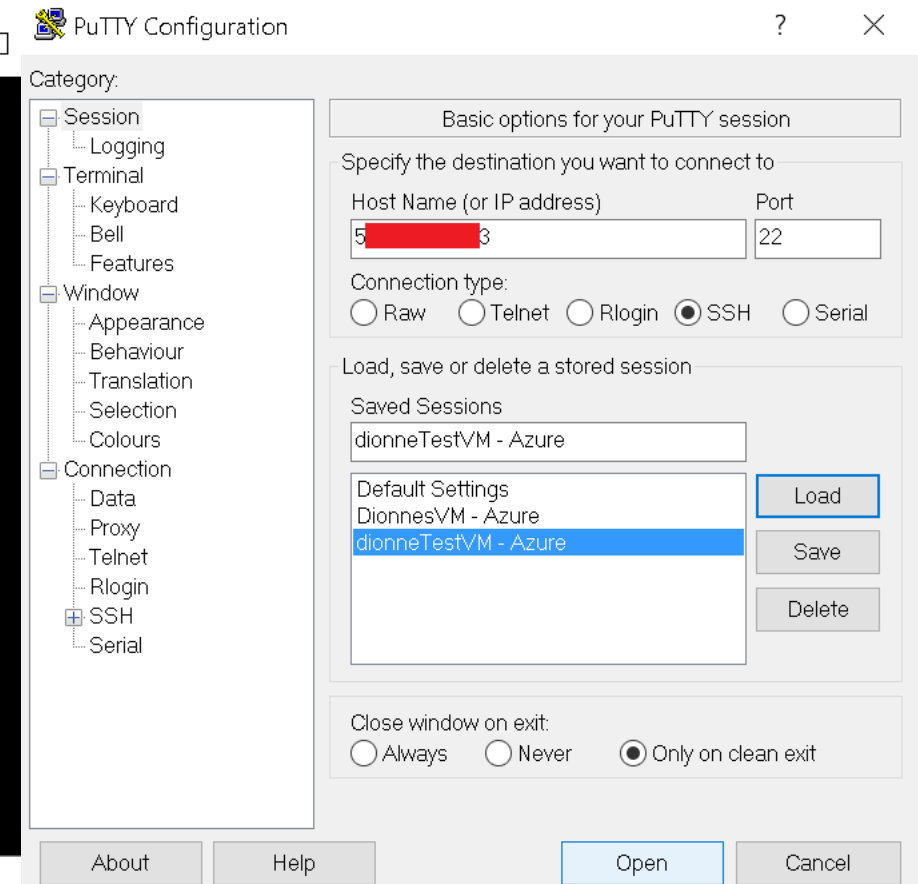
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

Last login: Sat May  6 00:00:18 2017 from [REDACTED]
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

dcfuser@dionneTestVM:~$
```

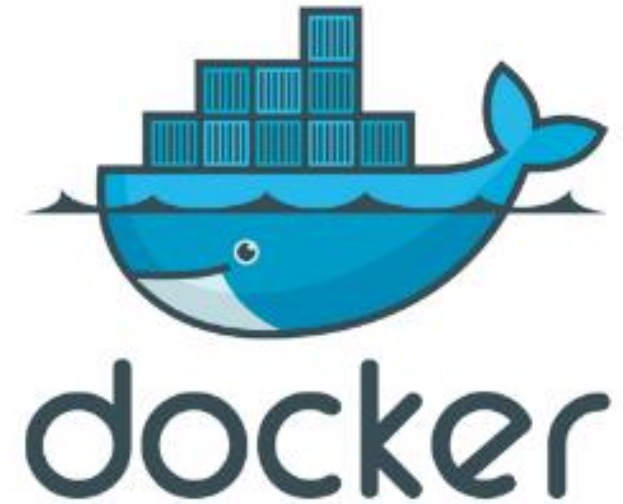


# AZURE LINUX VM - SET UP

*“Docker automates the repetitive tasks of setting up and configuring development environments so that developers can focus on what matters: building great software.*

*Developers using Docker don’t have to install and configure complex databases nor worry about switching between incompatible language toolchain versions. When an app is dockerized, that complexity is pushed into containers that are easily built, shared and run.”*

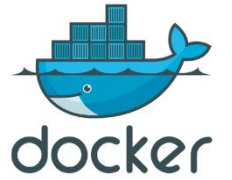
*[docker.com](https://docker.com)*





# AZURE VM AND DOCKER

## VIA DOCKER MACHINE AZURE DRIVER



```
C:\Users\Dionne>docker-machine create -d azure --azure-ssh-user ops --azure-subscription-id 7[REDACTED]9c --azure-resource-group dcfVMDockerMachineGroup --azure-location westeurope --azure-open-port 80 dcfmachine
```

```
Running pre-create checks...
(dcfmachine) Completed machine pre-create checks.
Creating machine...
(dcfmachine) Querying existing resource group. name="dcfVMDockerMachineGroup"
(dcfmachine) Creating resource group. name="dcfVMDockerMachineGroup" location="westeurope"
(dcfmachine) Configuring availability set. name="docker-machine"
(dcfmachine) Configuring network security group. location="westeurope" name="dcfmachine-firewall"
(dcfmachine) Querying if virtual network already exists. name="docker-machine-vnet" rg="dcfVMDockerMachineGroup" location="westeurope"
(dcfmachine) Creating virtual network. location="westeurope" name="docker-machine-vnet" rg="dcfVMDockerMachineGroup"
(dcfmachine) Configuring subnet. name="docker-machine" vnet="docker-machine-vnet" cidr="192.168.0.0/16"
(dcfmachine) Creating public IP address. static=false name="dcfmachine-ip"
(dcfmachine) Creating network interface. name="dcfmachine-nic"
(dcfmachine) Creating storage account. name="vhds4l0fa5lso94q2weg40qu" location="westeurope" sku=Standard_LRS
(dcfmachine) Creating virtual machine. location="westeurope" size="Standard_A2" username="ops" osImage="canonical:UbuntuServer:16.04.0-LTS:latest" name="dcfmachine"
Waiting for machine to be running, this may take a few minutes...
Detecting operating system of created instance...
Waiting for SSH to be available...
Detecting the provisioner...
Provisioning with ubuntu(systemd)...
Installing Docker...
Copying certs to the local machine directory...
Copying certs to the remote machine...
Setting Docker configuration on the remote daemon...
Checking connection to Docker...
Docker is up and running!
To see how to connect your Docker Client to the Docker Engine running on this virtual machine, run: docker-machine env dcfmachine
```

# AZURE VM AND DOCKER EXTENSION VIA ARM Templates

<https://portal.azure.com/#create/Microsoft.Template>

## Azure QuickStart Template

<https://github.com/Azure/azure-quickstart-templates/tree/master/docker-simple-on-Ubuntu/azuredeploy.json>

Edit template

Edit your Azure Resource Manager template

+

Add resource

↑

Quickstart template

↶

Load file

⬇

Download

Parameters (5)

Variables (17)

Resources (6)

[parameters('newStorageAccountName')]

[variables('publicIPAddressName')]

[variables('virtualNetworkName')] (...)

[variables('nicName')] (Microsoft.Ne...

[variables('vmName')] (Microsoft.C...

[concat(variables('vmName'), '/', var...

```
1 {
2   "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTe
3   "contentVersion": "1.0.0.0",
4   "parameters": {
5     "newStorageAccountName": {
6       "type": "string",
7       "metadata": {
8         "description": "Unique DNS Name for the Storage Account where the \
will be placed."
9       }
10    },
11    "adminUsername": {
12      "type": "string",
13      "metadata": {
14        "description": "Username for the Virtual Machine."
```

Save

Discard

Deploy an Ubuntu VM with Docker Engine

Azure quickstart template

TEMPLATE

docker-simple-on-ubuntu

6 resources

Edit template

Learn more

BASICS

\* Subscription

Visual Studio Professional

\* Resource group ⓘ

Create new

Use existing

dcfVMDockerExtensionGroup

\* Location

West Europe

SETTINGS

\* New Storage Account Name ⓘ

dcfvmdockerextstorage01

\* Admin Username ⓘ

dcfvmdockerextuser

\* Admin Password ⓘ

.....

\* Dns Name For Public IP ⓘ

dcfvmdockerext

Ubuntu OS Version ⓘ

16.04.0-LTS

TERMS AND CONDITIONS

Template information

Azure Marketplace Terms

Azure Marketplace

By clicking "Purchase," I (a) agree to the applicable legal terms associated with the offering; (b) authorize Microsoft to charge or bill my current payment method for the fees associated the offering(s), including applicable taxes, with the same billing frequency as my Azure subscription, until I discontinue use of the offering(s); and (c) agree that, if the deployment involves 3rd party offerings, Microsoft may share my contact information and other details of such deployment with the publisher of that offering.

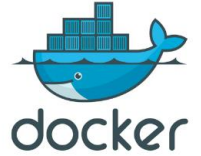
Microsoft assumes no responsibility for operations performed by third-party templates and does not provide rights for third...

☐ I agree to the terms and conditions stated above

☐ Pin to dashboard

Purchase

# DOCKER EXAMPLE



- **Pre-built Image - Oracle XE 11g** <https://hub.docker.com/r/wnameless/oracle-xe-11g>

```
docker pull wnameless/oracle-xe-11g
```

```
docker run -d -p 49160:22 -p 49161:1521 -p 49162:8080 wnameless/oracle-xe-11g
```

- **Custom Image – Based on JBOSS Wildfly** <https://hub.docker.com/r/jboss/wildfly>

1) Create a Dockerfile

```
FROM jboss/wildfly
```

```
ADD hello.war /opt/wildfly/standalone/deployments/
```

2) Build the Dockerfile to create an image: ***docker build --tag=wildfly-app .***

3) Run container based on the built images: ***docker run -it -p 8080:8080 wildfly-app***

4) Check container is running: ***docker ps***



~\$vi docker-compose.yml

# DOCKER EXAMPLE VIA COMPOSE

Create and start containers

which containers are running

~\$**docker-compose up -d**    ~\$**docker-compose ps**

```
version: '3'
services:
  springwebapp:
    build:
      context: .
      dockerfile: springwebapp.Dockerfile
    image: dionnecf/springwebapp
    ports:
      - "8080:8080"
    networks:
      - net-todo
    volumes:
      - .:/vol/development
    depends_on:
      - dcfmysqlserver
    container_name: springwebapp
```

```
$ docker-compose ps
```

Name	Command	State	Ports
dcfmysqlserver	/entrypoint.sh mysqld	Up	0.0.0.0:3306->3306/tcp, 33060/tcp
springwebapp	mvn clean package exec:java	Up	0.0.0.0:8080->8080/tcp

```
dcfmysqlserver:
  build:
    context: .
    dockerfile: springmysqldb.Dockerfile
  image: dionnecf/dcfmysqlserver
  ports:
    - "3306:3306"
  networks:
    - net-todo
  environment:
    MYSQL_DATABASE: tododb
    MYSQL_USER: dcfuser
    MYSQL_PASSWORD: dcfpassword
    MYSQL_ROOT_PASSWORD: rootpassword
  container_name: dcfmysqlserver
```

Check webapp works

```
$ curl -H "Content-Type: application/json" -X GET "http://192.168.99.100:8080/"
<h1>Hi, Welcome to my ToDoList app running in Docker with Mysql... :)</h1>
```

Check webapp retrieves data from database

```
$ curl -H "Content-Type: application/json" -X GET "http://192.168.99.100:8080/todos"
[{"id":1,"itemName":"Train Travel","itemDesc":"Remember to purchase your train ticket","createdDate":1491436800,"completed":false}, {"id":3,"itemName":"Snacks","itemDesc":"Remember to purchase some snacks before you go","createdDate":1491436800000,"completed":false}]
```

Tail the logs for a container

~\$**docker-compose logs springwebapp -f**

```
networks:
  net-todo:
    driver: bridge
```

# DOCKFILE FOR CUSTOM IMAGES

~\$ vi springwebapp.Dockerfile

```
FROM java:8

MAINTAINER Dee Farrell <dee@dcfhosting.co.uk>

# update packages and install maven
RUN \
    export DEBIAN_FRONTEND=noninteractive && \
    sed -i 's/# \(..*multiverse$\)/\1/g' /etc/apt/sources.list && \
    apt-get update && \
    apt-get -y upgrade && \
    apt-get install -y vim wget curl maven

# attach volumes
VOLUME /vol/development

# create working directory
RUN mkdir -p /vol/development
WORKDIR /vol/development

# maven exec
CMD ["mvn", "clean", "package", "exec:java"]
```

~\$ vi springdbserver.Dockerfile

```
FROM mysql/mysql-server

MAINTAINER Dee Farrell <dee@dcfhosting.co.uk>

ADD data/ /docker-entrypoint-initdb.d/
```

## Life Before Docker



Stressed Out!

## Life Since Docker



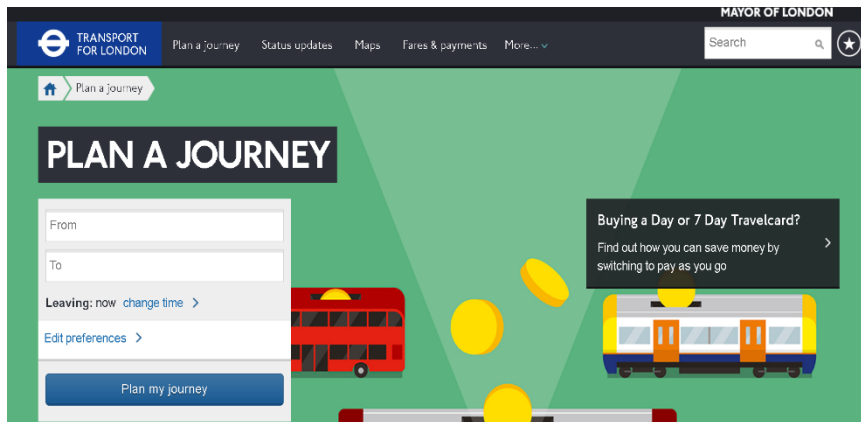
Stress Free and Relaxed

# TFL AND THE CLOUD



**AZURE**

**AWS**



**AWS**

**AZURE**



# TFL AND THE CLOUD

---



- TfL Wide Application Hosting Strategy currently being planned
- Cloud First approach, followed by Hybrid Cloud/On-premises approach
- Initial review shows over 50% of existing applications ready for cloud hosting (i.e. lift and shift).
- Deeper dive discovery is planned to confirm.



# JOURNEY TO THE CLOUD

## A Developer's Perspective

- Cloud Computing provides time savings, simplicity, flexibility for Developers
- Many Cloud Computing Providers out there to learn about
- Get started with Azure App Service and Azure Virtual Machines
- Save your sanity, just use Azure VMs with Docker and Compose
- Give it a go.... Cloud computing is not as complicated as it first seems

<https://azure.microsoft.com/en-us/try/app-service/>

<https://azure.microsoft.com/en-gb/campaigns/developer-guide/>