This document provides a walkthrough of the process required to deploy a Web application to the Azure platform

It will assume that the reader already has access to an Azure account and is on the (new) Azure portal page. To set up an account, please visit <https://manage.windowsazure.com/> for further instruction.

# Create container registry

1. Click '+'
2. Search 'container registry' (Azure Container Registry)
3. Click 'Azure Container Registry'
4. Click 'create'
5. Complete all fields of the form as the following:-
   1. Enter Registry Name (e.g. <yourapp>container)
   2. Ensure correct subscription selected (Free Trail)
   3. Resource Group: add new or select existing
   4. Enter Location (UK South)
   5. Select Admin User (Enabled)

# Obtain access keys

1. Access azure credentials (at time of writing, it was necessary to log out and back in again to retrieve the expected information)
2. Click 'all resources'
3. Click <yourapp>container listing
4. Click 'access keys'
5. Record login server (URI), username, password

# Install Vagrant-Azure plugin

Once a container has been created on the Azure account and the user credentials have been obtained, it is necessary to ensure that the C++ compiler is installed as a prerequisite to the installation of the Vagrant-Azure plugin.

1. Open a terminal window and at the (root) prompt type:-

> sudo yum install gcc-c++

1. Next, install the Vagrant-Azure plugin

> vagrant plugin install vagrant-azure

# Create Azure management certificate

It is now necessary to create an X509 key-pair in order to authenticate the client to the hosting Azure server.

1. Open a terminal window and at the (root) prompt type:-

> openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout ~/.ssh/azurevagrant.key -out

~/.ssh/azurevagrant.key

> chmod 600 ~/.ssh/azurevagrant.key

> openssl x509 -inform pem -in ~/.ssh/azurevagrant.key -outform der -out ~/.ssh/azurevagrant.cer

# Upload Azure management certificate

1. Log into Azure Classic Portal (<https://manage.windowsazure.com/>)
2. Ensure the correct subscription is selected (Free Trial at time of writing)
3. Navigate Settings listed in the left (scrollable) menu panel
4. Navigate Management Certificates
5. Click 'Upload'

# Build containers

The application used in this example for demonstration purposes comprises of two components that will need to be packaged as a container using the Docker utility.

1. MongoDB

> docker fetch <container>

> docker tag mongo <URI>/<container>

> docker push (<URI>/<container>)

1. Application (MEAN)

> npm install

> npm run ng build --prod

> docker build ./ -t <URI>/<container>

> docker login -u <username> -p <password> <server>

> docker push (<URI>/<container>)

# Provision Vagrant box

1. Set the azure credentials recorded before as environmental variables from a terminal window

> export DOCKER\_USER=<azure\_container\_user>

> export DOCKER\_PASSWORD=<azure\_container\_password>

> export DOCKER\_REGISTRY=<azure\_container\_registry\_name>

2. Set the azure credentials recorded before as environmental variables from a terminal

> vagrant up