AKS PROJECT TO INSTALL HEML and azure CL on command prompt admin.

**Requirements**

* Windows 7+ / Windows Server 2003+
* PowerShell v2+
* .NET Framework 4+ (the installation will attempt to install .NET 4.0 if you do not have it installed)

That's it! All you need is choco.exe (that you get from the installation scripts) and you are good to go! No Visual Studio required.

**Basic Chocolatey Install**

Chocolatey installs in seconds. You are just a few steps from running choco right now!

1. First, ensure that you are using an [***administrative shell***](https://www.howtogeek.com/194041/how-to-open-the-command-prompt-as-administrator-in-windows-8.1/) - you can also install as a non-admin, check out [Non-Administrative Installation](https://docs.chocolatey.org/en-us/choco/setup#non-administrative-install).
2. Copy the text specific to your command shell below.
3. Install with cmd.exe

Run the following command:

|  |
| --- |
| @"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -InputFormat None -ExecutionPolicy Bypass -Command "iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))" && SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin" |

1. Paste the copied text into your shell and press Enter.
2. Wait a few seconds for the command to complete.
3. If you don't see any errors, you are ready to use Chocolatey! Type **choco** or **choco -?** now, or see [Getting Started](https://docs.chocolatey.org/en-us/getting-started) for usage instructions.

INSTALL HEML via chocolatey.

|  |
| --- |
| choco install kubernetes-helm |

Login to portal via command line

|  |
| --- |
| az login |

Create a resource group

|  |
| --- |
| az group create –name AKSRG –location uksouth |

Create ACR and specify the sku

|  |
| --- |
| az acr create --resource-group AKSRG --name radwanacr --sku Basic |

Create AKS CLUSTER and attach to ACR at once

|  |
| --- |
| az aks create –g AKSRG –n MyAKS --location uksouth –attach-acr radwanacr --generate-ssh-keys |

Check if “kubectl” is installed. If not, simply run;

|  |
| --- |
| az aks install-cli |

Request permission to access kubernetes cluster. THESE credentials get stored in the C:\Users\USER\.kube\config file

|  |
| --- |
| az aks get-credentials --resource-group AKSRG --name MyAKS |

|  |
| --- |
|  |

Sample GitHub js app to be used for this project

|  |
| --- |
| https://github.com/Azure/dev-spaces/ |
| https://github.com/Azure/dev-spaces/samples/nodejs/getting-started/webfrontend |

Since we using command line for this project, check if git is installed. If not, install git and clone that repo. You can install git using the choco package manager.

|  |
| --- |
| choco install git |

Clone the repo

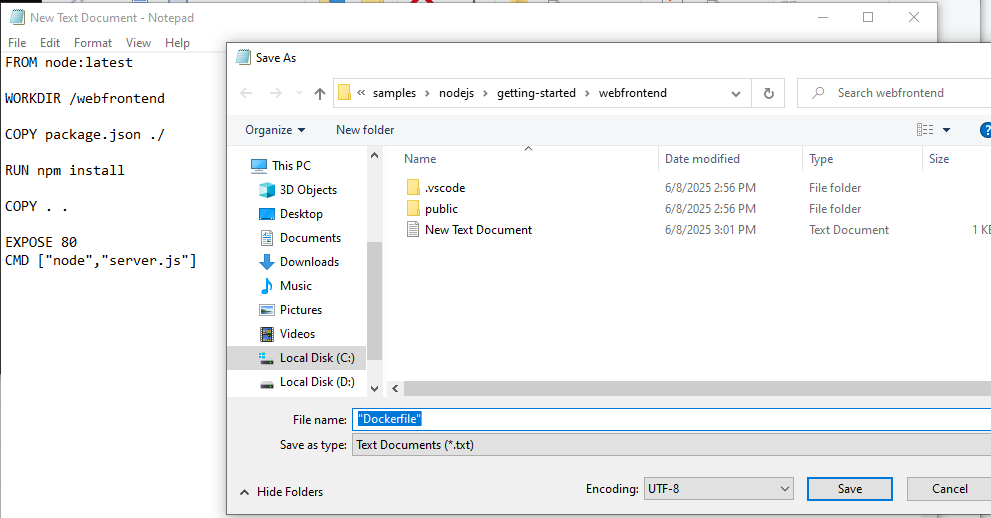
|  |
| --- |
| git clone https://github.com/Azure/dev-spaces/ |

Navigate to the Webfrontend directory

Create a dockerfile and type the command below

|  |
| --- |
| FROM node:latest  WORKDIR /webfrontend  COPY package.json ./  RUN npm install  COPY . .  EXPOSE 80  CMD ["node","server.js"] |

Save it with a filename “Dockerfile”



|  |
| --- |
|  |

Build that image and push it to the ACR via the command line.

|  |
| --- |
| az acr build --image webfrontend:v1 --registry radwanacr --file Dockerfile . |

Once this is done, go the webportal and verify the changes in your ACR repository

In your default path on the cmd terminal, create a helm templatecalled “webfrontend”

|  |
| --- |
| helm create webfrontend |

A complete helm directory and yaml files are going to get created.

Open the **values.yaml** file

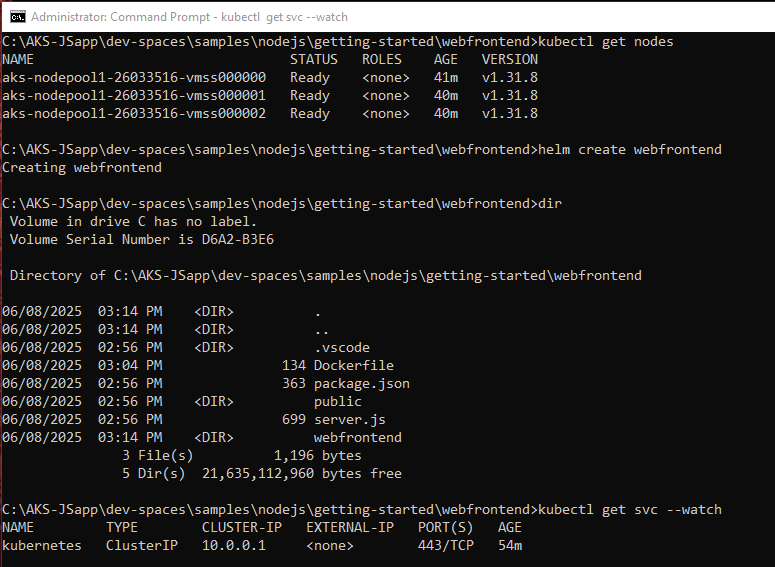
Change the **repository** value to that of your acr: **<reponame>.azurecr.io/imagename**

Change **service type** from **clusterIP** to **LoadBalancer**

**Save the values.yaml file and exit.**

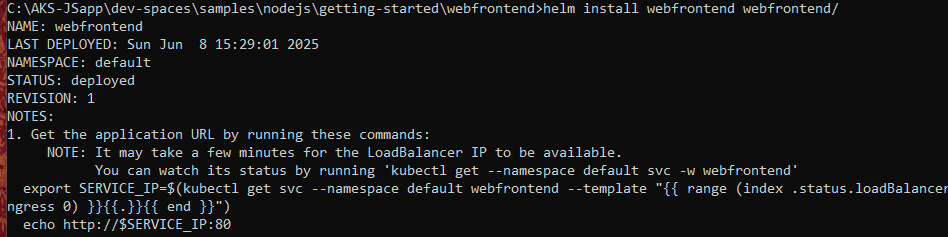
Watch whats going on to services in your cluster before deploying

|  |
| --- |
| kubectl get svc --watch |

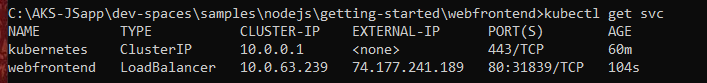


Launch the **helm install command**

|  |  |
| --- | --- |
| **Default helm syntax** | **helm install <release-name> <chart> [flags]** |
| **Run the following** | **helm install webfrontend webfrontend/** |

****

**$ kubectl get svc**

****

Firstly, browse to the k8s dashboard using cmd terminal

|  |
| --- |
| $ az aks browse --resource-group AKSRG --name MyAKS |

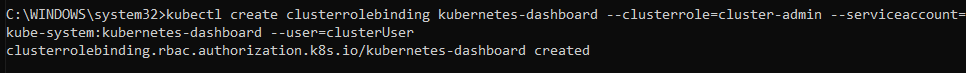
Before the command above can work, you would have to give the permission to the cluster user

Delete the cluster binding (if a prev one exist)

|  |
| --- |
| $ kubectl delete clusterrolebinding kubernetes-dashboard |

Create a cluster role binding

|  |
| --- |
| $ kubectl create clusterrolebinding kubernetes-dashboard --clusterrole=cluster-admin --serviceaccount=kube-system:kubernetes-dashboard --user=clusterUser |



The user created above is going to be used to access the kubernetes dashboard.

OPTIONAL: IF the command above fails and says “kubernetes-dashboard” already exist, then delete the previous credentials and recreate a new credential afterwards.

|  |
| --- |
| $ kubectl delete clusterrolebinding kubernetes-dashboard |

THEN, browse to the kubernetes dashboard.

|  |
| --- |
| $ az aks browse --resource-group AKSRG --name MyAKS |

NB: AZURE DASHBOARD IS NOT DEPRECATED DUE TO SECURITY CONCERNS

Output of my app from its loadbalancer IP

