SOLABORATE ASSIGNMENT

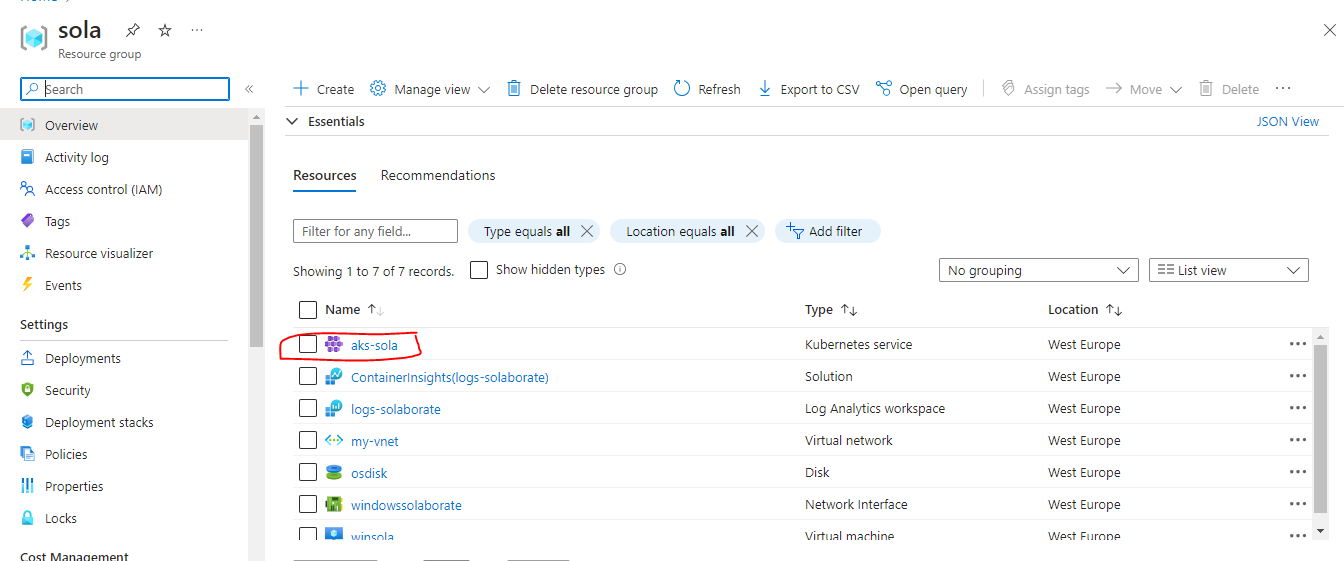
**All source code for the assignment could be found on the following Git repository:** [**https://github.com/vetonnshala/sola**](https://github.com/vetonnshala/sola)

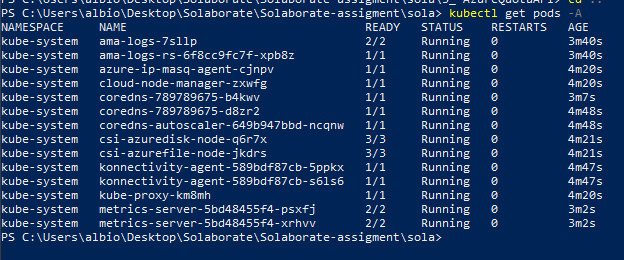
**For the assignment I used Azure as a cloud provider.**

## Create k8s Cluster

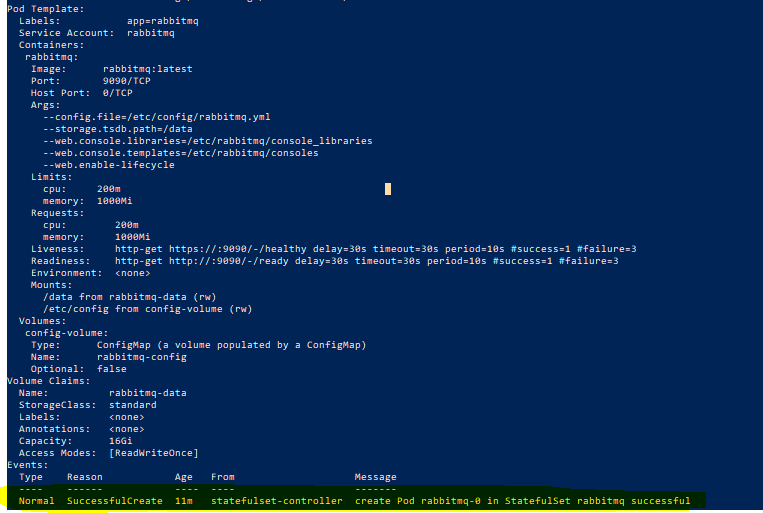
On the 1\_k8s-Cluster folder, I have created terraform scripts that create the required infrastructure of the Cluster with a stateful service deployed on it and Windows virtual machine with AD enabled.

These terraform scripts create a k8s cluster on Azure AKS service together with a windows VM which has AD enabled as an *azurerm\_virtual\_machine\_extension* terraform resource. On this AKS cluster I have deployed RabbitMQ as state-full service as a terraform resource which can be found on *stateful\_set\_rabbitmq.tf* terraform script. Also there I have applied necessary configurations such as **update\_strategy** that support zero-down time during service upgrade.

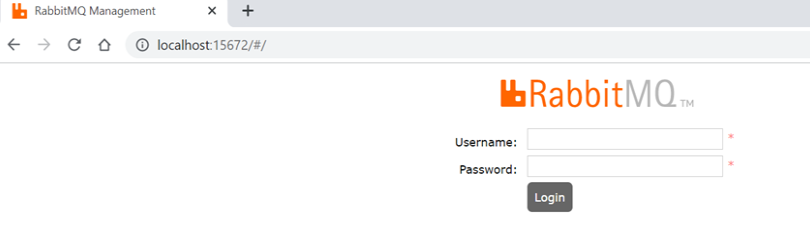




RabbitMQ deployment







## Scripting

As seen on 2\_Scripting project folder, I created solaboratebash.sh script which can do different operations such as:

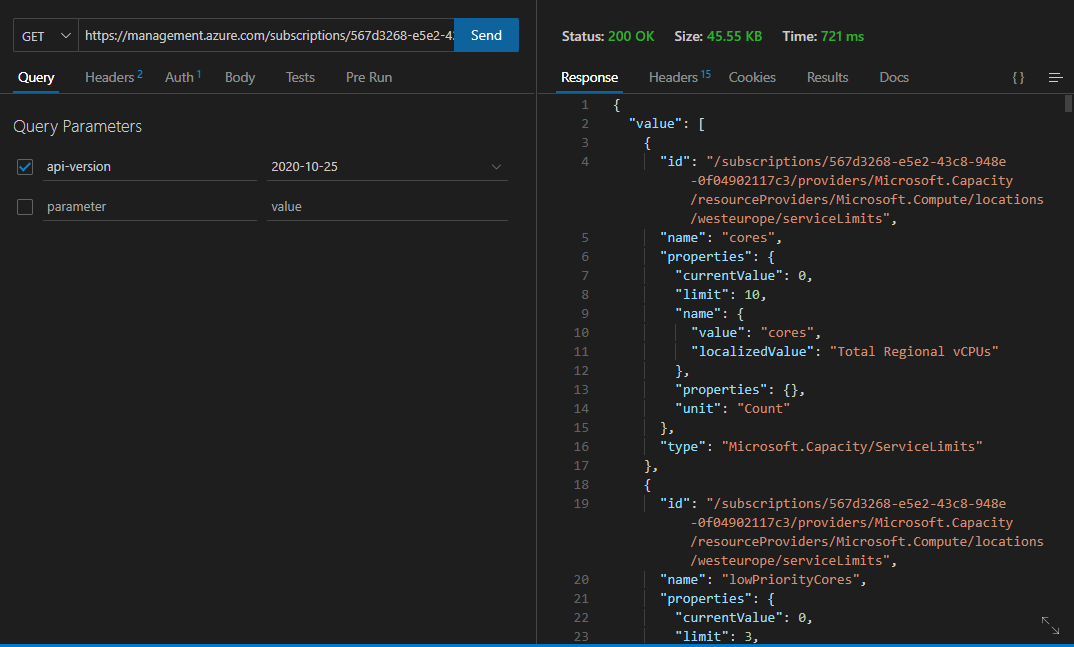
* Write text to a file
* Copy content of one file to another
* Filter content of a file
* Count special characters
* Uppercase the content and save to a new file
* Calculate the average word length
* Redirect the output to a file

Please also check the different text files that are being used by the bash script.

## Use Azure Quota API to retrieve list of quota limits for subscription or scope.

Using Thunder Client extension on Visual Studio Code, I sent the request to Azure to retrieve the quota limits for subscriptions. I used a Bearer token to authenticate with Azure and created the PowerShell script which can be run to retrieve the results afterwards. I also needed to register Microsoft.Capacity resource provider on my Azure account.

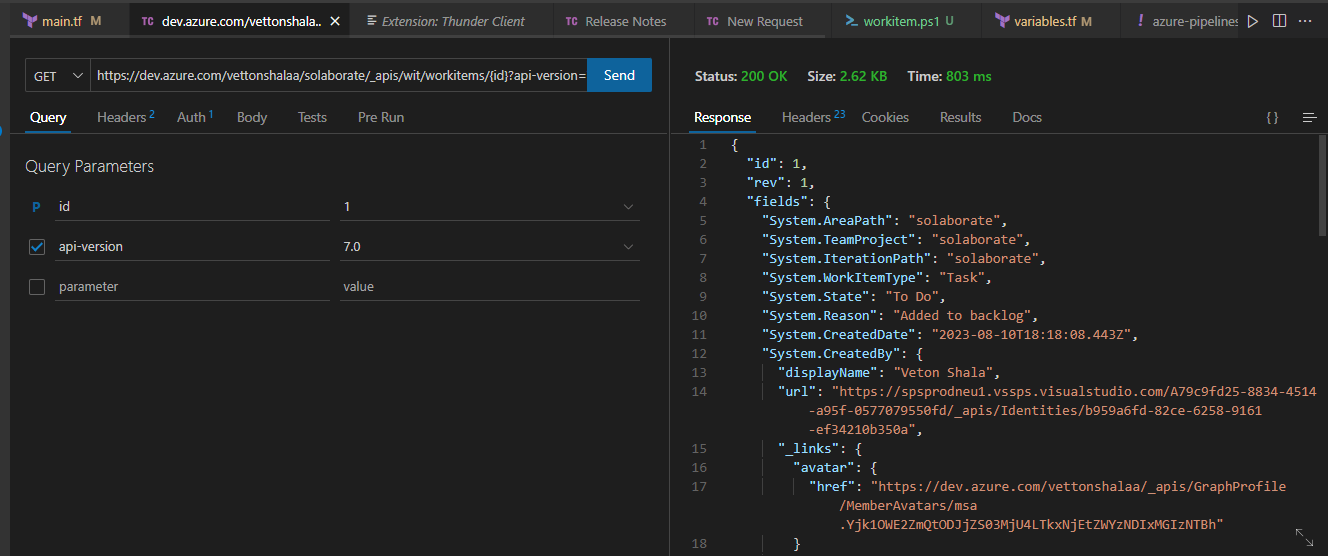
Get *https://management.azure.com/subscriptions/567d3268-e5e2-43c8-948e-0f04902117c3/providers/Microsoft.Capacity/resourceProviders/Microsoft.Compute/locations/westeurope/serviceLimits?api-version=2020-10-25*



PowerShell scrip can be found on the *3\_ AzureQuotaAPI* project folder.

## Use Azure DevOps REST API to list Work Items type 'Bug'.

Same as on the third assignment, I used Thunder Client extension on VS Code to retrieve the needed results, also created the PowerShell script which can be found on 4\_AzureWorkItems project folder. Here, I used PAT which I created on Azure DevOps to authenticate with Azure and retrieve work items of type Bug. Since I didn’t have any bug work item, I created a task on Azure DevOps and tested out the script which showed the task I created as shown on the screenshot below.



## Docker

I created a Dockefile which can be found on 5\_Docker project folder. This Dockerfile creates an Docker Image which as base layer will use ubuntu:20.4 and holds two arguments which pass down their value on ENV variables that I set such as User\_Name and App\_Verison. Then I set a step to install curl tool inside /app directory and another step where I copied a local text file inside this docker image.

