Analysis of PaaS & SaaS Interoperability through Azure, Red Hat OpenShift and AWS

Javedali Shaikh (16212373) | Sanjay Singh (16211668) | Kevin Shortall (16213216)
Team 10
06/12/2017

Introduction

The advent of the cloud industry has changed the way the world interacts with computing. Compared to traditional computing, cloud computing offers benefits such as increased cost efficiency, vastly increased storage potential, and backup and recovery. These advantages have seen most medium- and large-scale companies shift their businesses to the cloud. The resulting growth in the cloud computing industry has seen providers branch into an increasing number of areas, with many cloud providers now offering services such as data gathering and data analytics.

In this paper we will compare PaaS solutions on the two market leaders in commercial cloud (Amazon's AWS and Microsoft's Azure) and one open-source cloud platform (Red Hat OpenShift). We will deploy to the platforms an application written in asp.net programming language, designed for face detection in images. We will examine various aspects of the platforms in relation to this interaction, such as storage, auto-scaling, interoperability and backup/recovery.

Goals of the Project

- Exploring PaaS (website hosting/deploying and storage) in all 3 cloud providers
- Cloud Interoperability using SaaS
- Migration of ASP.NET based website from one cloud to another (Interoperability)
- Scalability - Auto Scaling

Method

Due to the expertise within the team, it was decided that Microsoft Azure would be the anchor platform, with AWS and Red Hat OpenShift working in comparison to Azure. The code for the application was built using asp.net language. The method consisted of firstly deploying a simple Face Detection app on all platforms that allowed one image to be uploaded at a time. Once successful, the code was updated to retrieve multiple images from storage locations within the relevant cloud platforms. The code was also adapted to apply an artificial loading to the CPU to confirm that auto-scaling is working as planned.

Description of chosen Cloud Platforms

Microsoft Azure

Microsoft Azure aka Windows Azure is one of the biggest cloud provider for building, deploying and managing application and services. It provides all 3 major aspects of cloud IaaS, PaaS and SaaS. Released to world first in 2008 October, now Azure has market share ~11% [1].

Red Hat OpenShift

Red Hat OpenShift is an Open Source container based software development and management Hybrid Cloud Application Platform from Red Hat. As part of Open Source Software community, Red Hat OpenShift supports multiple languages and services, deployed in applications through container images. The Container or cartridge image can be web frameworks, monitoring services, databases or connectors to external backends. Due to container images wise packaging services and framework, administrators and developers can focus more on the delivery side with respect to code security and logic [3].

AWS

Amazon's AWS Cloud Platform is the longest established major cloud platform in the market and has continuously dominated the market share, with reports [2] in November 2017 of a 44% public cloud market share. AWS continues to offer customers the largest range of cloud projects, with recent branching into data analytics and machine learning, and reported future ventures into healthcare and education.

Website Implemented and deployed in 3 clouds

We created an ASP.NET MVC 4.5 website for recognizing Faces. The code was 50% taken from MSDN [4] and rest 50% was added by us.

The main features of this website are - **Face Detection, Multi Face Detection and Auto Scale trigger.**

Azure url: http://face-api.azurewebsites.net/FaceDetection

AWS url: http://faceapimvcweb-dev.eu-west-1.elasticbeanstalk.com/

Red Hat OpenShift url: http://sampleapp-projectcloud.193b.starter-ca-central-1.openshiftapps.com/
http://sampleapp-projectcloud.193b.starter-ca-central-1.openshiftapps.com/

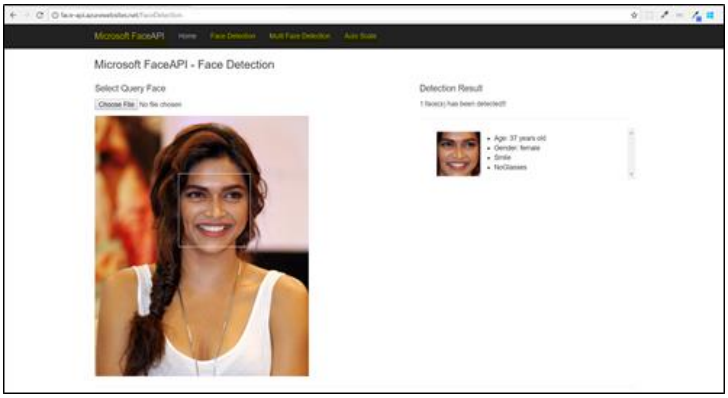

Figure 1 - Deployed Web App interface

Project Diagram

Below picture represents the Face API hosted in 3 different cloud solutions. It shows interoperability example we implemented between clouds and also interoperability of moving website from one cloud to another. In the later, as we can see in the diagram, Red Hat OpenShift failed to host the ASP.NET MVC 4.5 based website due to lack of support. Lastly, it depicts the use of PaaS and SaaS in our project.

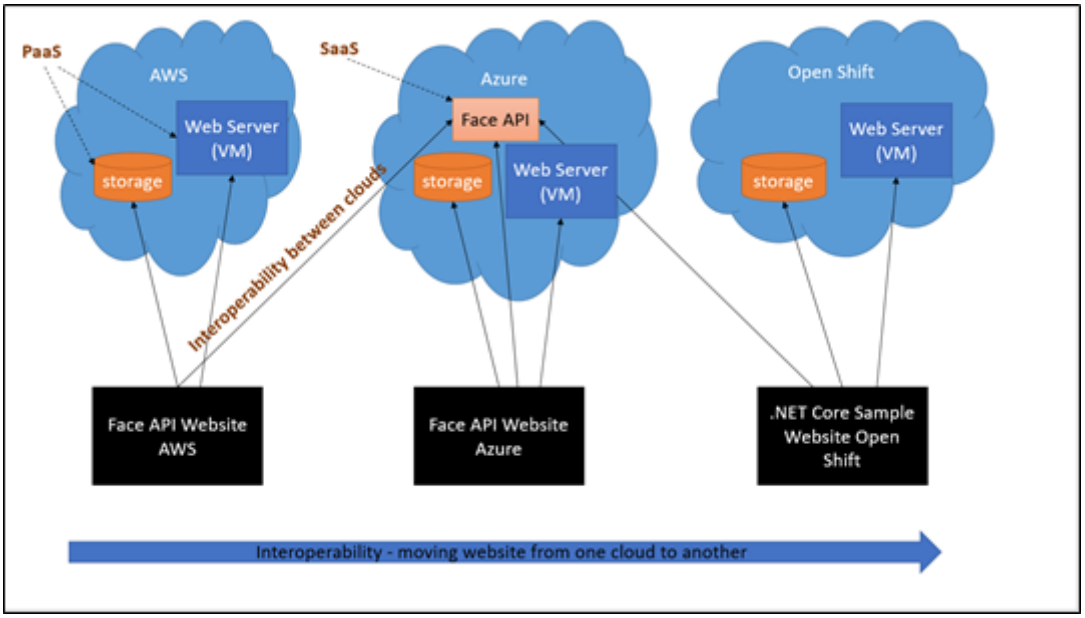

Figure 2 - Project Diagram

Summary Comparison Table

(Weight 0 -5, Marking 0 -5)

	Weight	Azure	Red Hat OpenShift	AWS
Creating an Account / Installation	2	4	4	4
Documentation & Community	3	4	2	3.5
Management Portal	3	5	3	3.5
Storage	4	5	2.5	4
Deploying Applications	3	5	3	3.5
Interoperability	4	5	2.5	5
Costs & Free Plans	4	3	3	4
Configurations	3	4.5	3	5
Backup resources	3	4.5	2.5	5
Scalability	5	5	3	5
Ease of Use/User-Friendliness	3	4.5	2	3
Monitoring	3	4	3	4.5
Security	N/A	N/A	N/A	N/A
Total Score	**200**	179.5	110.5	169

Comparison points

Creating an Account / Installation

Microsoft Azure (4/5)

Offers a free trial account creation, with microsoft, hotmail, outlook or any other microsoft Id. The trial account is valid for 12 months (1 year) and has a credit of $200. We can use many of the services provided by Azure using this money. We bought a **Standard size VM** were our website was hosted. The VM specification are - 1 Core, 1.5 GB of RAM, 50 GB storage and upto 10 auto scale instances. Also, we bought Blob storage which we used to host images for our website and also to create backup for our website. The account creation process was quick, easy and also very beneficial as the account is valid for 12 months.

Red Hat OpenShift (4/5)

Has free signoff using existing Google, GitHub, Linkedin, Microsoft or RED HAT account. The free subscription is valid 90 days with limited resource i.e 1GB of memory, 1GB of persistent storage, 1GB of terminating memory included with online starter. Additional memory is not available for this plan. Also no credit card required for Starter online account.

AWS (4/5)

The creation of an AWS account is quite straightforward and intuitive. There is a free usage allowance given to each new user but credit card details must be supplied in case of exceeding those thresholds. There is no local installation required to work with AWS but the installation of Visual Studio 2017 (Community Version) was of assistance throughout this project.

Documentation & Community

Microsoft Azure (4/5)

Has a huge documentation repository with hundreds of examples for multiple languages. E.g. there are examples on how to access the storages in .NET, JAVA, python, Javascript etc. languages. This gives benefit that diverse set of developers can use Azure efficiently and no one is bound to one programming language. We found plenty and detailed documentation especially on creation of App Services (Web Apps), storages (blob, file, table etc.) which was very useful for our project. Azure almost stand second to AWS, and also has great community support. We can ask question on various forums like - reddit, stackoverflow, MSDN forums, Azure Feedback and many more [5].

Red Hat OpenShift (2/5)

Red Hat OpenShift has very limited documentation and support. It was very difficult to deploy to faceapi code as the required template was not available. Also no contact details for add-on and when contact there is no response.

AWS (3.5/5)

Has a plethora of information, support and documentation, throughout the internet and within the AWS environment. This information ranges from personal blogs to forums to online ‘how-to’ videos. While this is obviously an advantage in most scenarios, as a beginner with AWS and cloud in general, the breadth of information can make it difficult to find solutions to particular problems. Also, due to AWS’s comparative longevity in the cloud market, quite a sizeable amount of that information is now already outdated, such is the speed of evolution in the Cloud market. Indeed, one noticeable flaw in AWS’s documentation was how many of the ‘support’ pages were referencing old versions of the AWS GUI, which sometimes made it difficult to find certain aspects that were referred to.

Management Portal & Account Administration

Microsoft Azure (5/5)

Management portal for Azure is one of the best portal we have come across. Azure released a new portal this year which is more user friendly, efficient and gives user ability to do multiple things simultaneously. https://portal.azure.com/.

Top features of Azure Portal which we explored:

Dashboard: Users can create customized dashboard, where we can pin literally anything which is present in Azure like websites, storage etc. We used the dashboard to pin the Face API website, Blob storage, resources, **Http 500 Errors** which is really cool. Sample screenshot below:

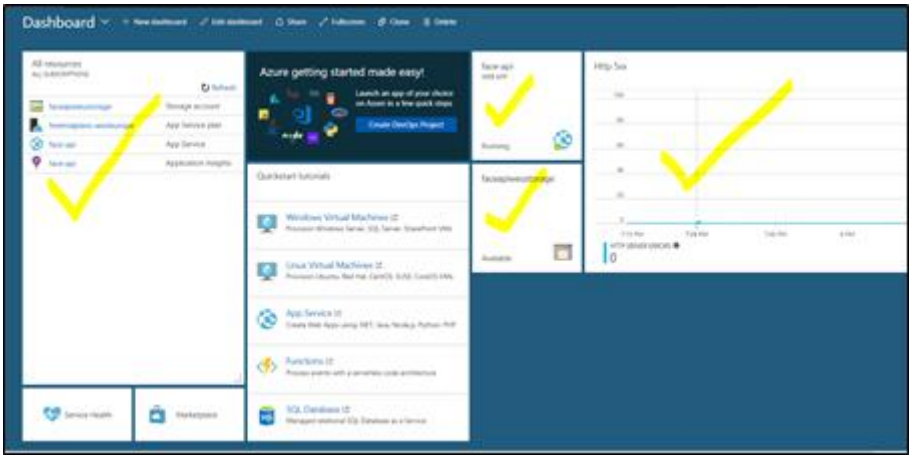

Figure 3 - Azure Dashboard

Multi-Column Layout: This layout gives a snapshot of all action a user takes to achieve one particular task. For e.g. in order to create a storage account, we need to do few subtasks. User can see all sub-tasks and also can go to any of those tasks at any time. Sample screenshot of configuring auto scale:

Notifications: Central location where all activities are notified like creation of web site, storage, auto scale rule, credit information and many more.

Lastly, the new portal is a Single Page Application giving user very high performance and fast UX.

Microsoft Azure has good administration section as well in the portal, where users can check the current spent money and also see the forecast of how the money will be used in next few days/weeks.

Red Hat OpenShift (3/5)

OpenShift has limited management console. There are the following options for Build, Configure and manage applications: Overview, Application, Builds, Resources, Storage and Monitoring.

Pods are the rough equivalent of a machine instance (physical or virtual) to a container. Each pod is allocated its own internal IP address, therefore owning its entire port space, and containers within pods can share their local storage and networking.

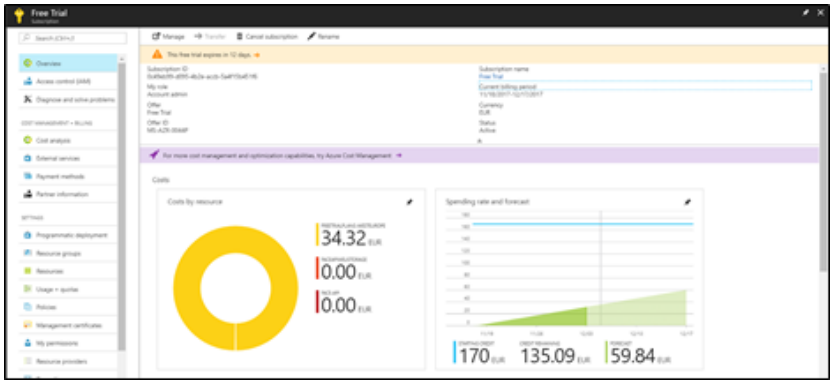

Figure 4 - Azure Cost Management Dashboard

AWS (3.5/5)

Administering your AWS account from a financial perspective is easy and straightforward thanks to the AWS Billing & Cost Management Dashboard which is available from the console banner. It gives a high-level breakdown of costs incurred, your current balance, and the option to delve deeper into the analytics of where and when that balance has been spent.

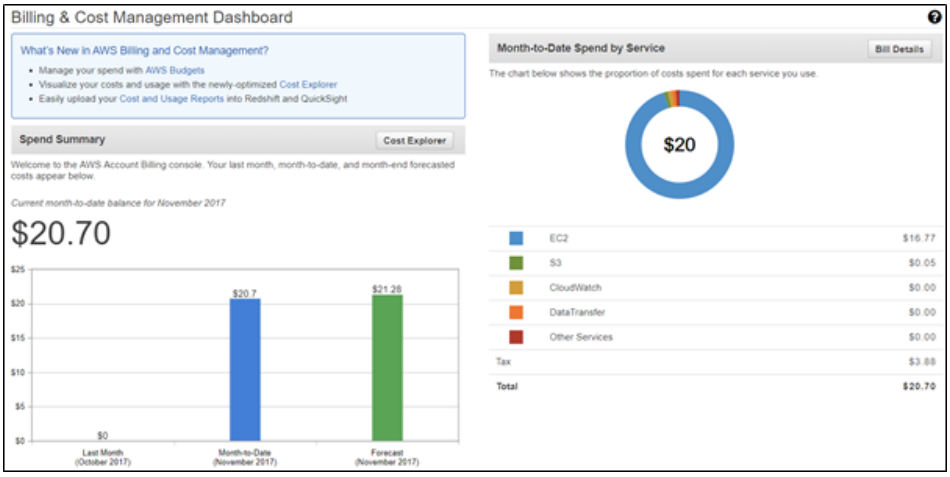

Figure 5 - AWS Billing Dashboard

There are numerous services within AWS and each has their own high-level dashboard. These are all easily accessible but, particularly as a beginner, it can be difficult to keep track of where certain information is accessible amidst so many services. All the information is in there somewhere, but navigation is far from foolproof and is not always intuitive.

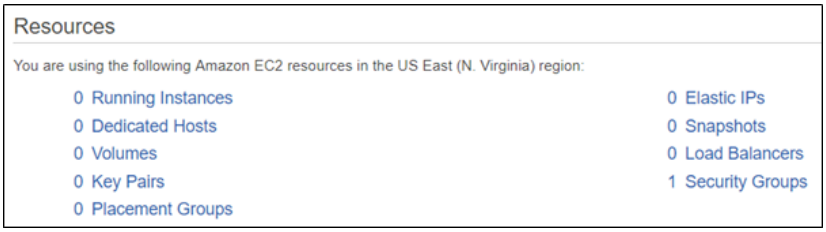

Figure 6 - AWS EC2 Dashboard

Public/Private/Enterprise options

All 3 cloud providers (Azure, Red Hat OpenShift and AWS) provides all 3 types of cloud option (public, private and enterprise). Exploring in detail these option was beyond the scope of this project. We are giving 5 marks to all 3, as at high level all 3 provide these options.

Data Storage

Microsoft Azure (5/5)

Provides tons of storage options - Blob, file, table, queue, SQL, Azure File Sync, Data Lake Store, Document DB and many more. We were interested in the classic type of storages blob and file. After few analysis we decided to go with Azure Blob storage as blob is really performant for storing binary data like text documents, image etc. We created a blob storage in West Europe data center and create a container to host our test images. The storage had the replication setting of 'Read-access-geo-redundant RA-GRS'. We kept the storage public for easier access and avoiding writing complex code for authentication and authorization. We also used this storage to store the Face API backup.

Azure offers REST APIs for blob storage. We added code in C# to access the blob storage container images using the REST APIs. The code was easy to write and there were minimum hassle to make it working. We can also query the data using .NET libraries in C#. But REST APIs gives a consistent approach to access various storage within Azure and also in different cloud platforms i.e. we can write code once and have configurations for different API URLs.

Red Hat OpenShift (2.5/5)

Allow Persistent data storage which leverages the kubernetes Persistent Volume subsystem. Openshift provides an API for developer and administrators that abstracts details of how storage is provided instead of how it is consumed.

Red Hat OpenShift has different types of Persistent Volumes for different delivery models. e.g Red Hat OpenShift Enterprise supports NFS, HostPath. With Red Hat OpenShift Free subscription the storage quota was limited.

AWS (4/5)

Uses object storage called Simple Storage Solution (S3). In this project, initial trial runs stored the images locally on the laptop, but were then transferred to an S3 folder in AWS. The .net source code then retrieved them once requested. Amazon states a delivery target of *"99.999999999% durability"* with S3 and that it *"is the most supported platform available."* As the largest cloud provider, there is no cause to doubt these claims, however to this user, the S3 folder structure lacked a certain finesse and smoothness in navigation. AWS offers additional storage services that were beyond the use of this project such as Amazon Elastic Block Store (EBS), predominantly used for data analytics and log processing, and a long-term data archiving service known as Amazon Glacier.

Deploying Applications

Microsoft Azure (5/5)

Web site can be deployed in following ways:

1. Manually uploading the files/package via Azure portal
2. Using Visual Studio IDE (Right click -> publish)
3. Azure Command Line

4. Azure Rest APIs

We implemented the deployment via Visual Studio. Below is the main screens:

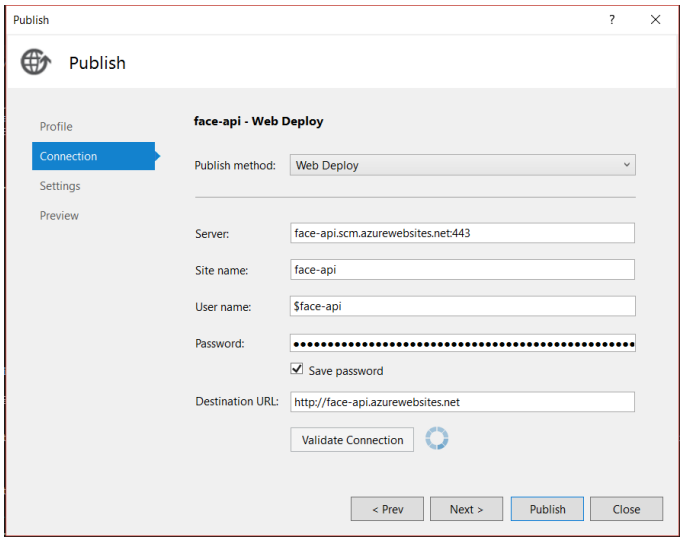

Figure 7 - Visual Studio Publish modal

Red Hat OpenShift (3/5)

Application can be deployed using online web console, command-line or (Visual Studio 20017) IDE. Once the application is created, the application code is pushed to Openshift online thru GitHUB.

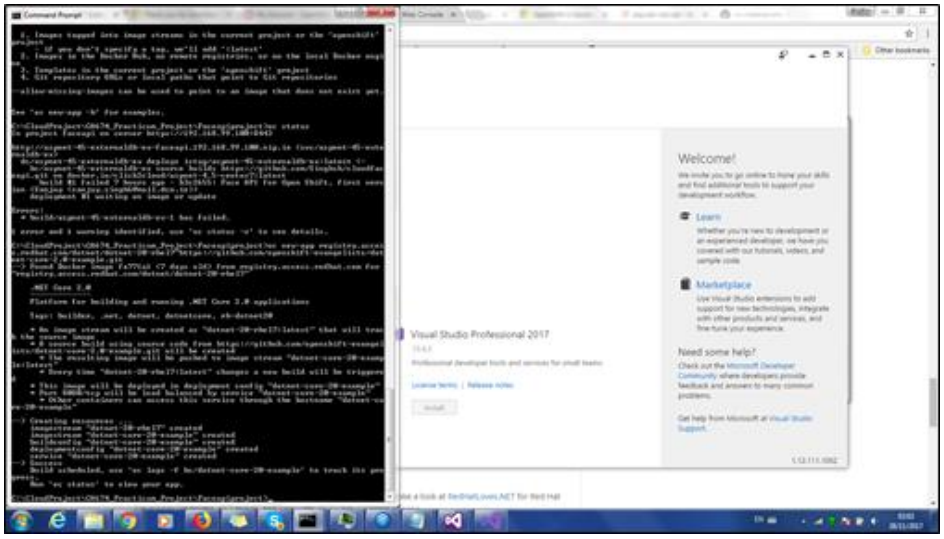

Figure 8 - Red Hat OpenShift

AWS (3.5/5)

From selecting the options in “Publish to AWS Elastic Beanstalk” in Visual Studio to having the environment deployed on AWS console, and accessible for interaction, took on average less than 2 minutes. The speed helps greatly with changing configuration, restarting apps or deploying, as the time required for experimenting or solving problems is much shorter.

Difficulties were experienced with regard to user permissions for newly deployed web apps which caused many delays before resolving. The details of this are covered in the Challenges section in this report. The problem appeared to be present by default, so is likely to be impacting many users.

Interoperability

Microsoft Azure (5/5)

As we first created the website using Microsoft technologies (ASP.NET), Azure didn't had any problems in deploying and interacting with the website.

Red Hat OpenShift (2.5/5)

Openshift does not support ASP.NET 4.5 MVC framework, hence unable to deploy FaceAPI website. However using third party Click2Cloud template, successfully deployed a sample webpage in ASP.NET 4.5 (not MVC).

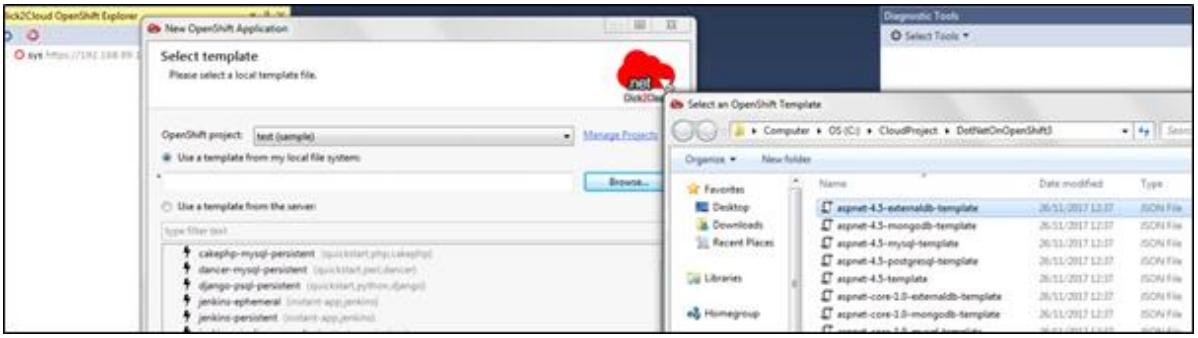

Figure 9 - Red Hat OpenShift

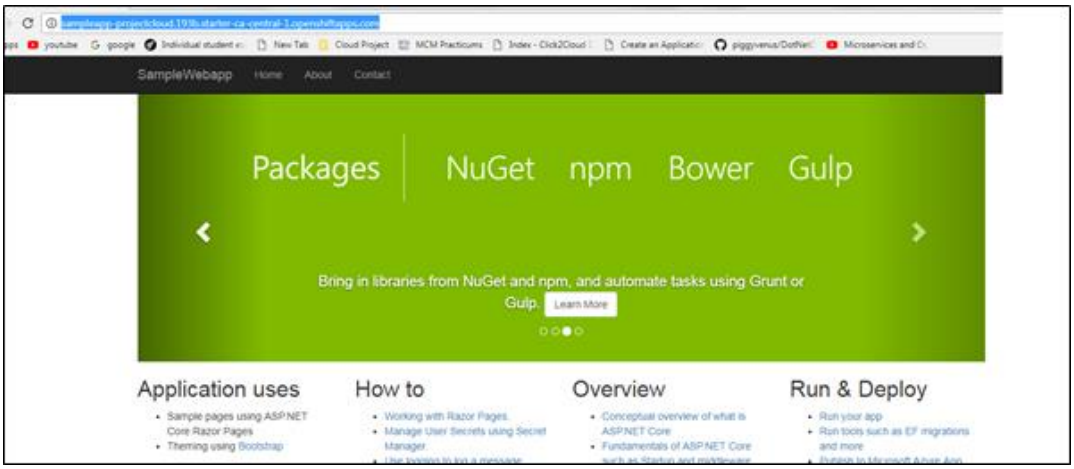

Figure 10 - .net Core Website hosted on Red Hat OpenShift

AWS (5/5)

As market leader, AWS does not have to worry about a lot of the interoperability problems from which many other providers suffer. Applications, services and platforms will all aim to be AWS-friendly in order to secure as many potential customers as possible. The addition of the AWS Toolkit add-on to Visual Studio is an example of this, and one of great help and convenience in this project. There were no interoperability issues of note experienced in this project.

Costs & Free Plans

Microsoft Azure (3/5)

Free trial account does not require any credit card information and is valid till 12 months. Apart from the free account we found that Azure is not cheap. Azure provides various tiers like Standard, Premium etc. which have different resource capabilities and features. We have used the Standard tier for website and storage. It will be difficult for startups and small scale companies to use Azure. Though they do offer shared hosting options which are very cheap, but raises the security concerns.

Red Hat OpenShift (3/5)

Online pricing is across two categories:

Starter (1 Project, 1GB Memory included, 1GB terminating memory**, 1GB storage included)

PRO-With Red Hat OpenShift PRO plan user can subscribe for just $50/month and additional memory for $0.0035/hour per GB billed monthly. For Professional projects and hosting upt 10 projects 2GB Memory included, 2GB up to 48GB Memory Available, Upto 1000 GB storage, Always on, Unlimited Usage, invite Collaborators to Projects, Support Custom Domains, Scheduled jobs [7].

Terminating Storage quota applies to pods with an active deadline. These pods usually include builds, deployers, and jobs.

AWS (4/5)

Requires credit card details at sign-up in case the 'Free Tier' usage thresholds are exceeded. Some tools within AWS, such as the CloudWatch monitoring service, are given as completely free usage ("Non-expiring offers"). These also include some of the newer offerings such as Amazon Mobile Analytics and AWS X-Ray.

Others are offered as free for a 12-month introductory period once certain usage thresholds are not exceeded. These include some of the more familiar AWS tools such as Elastic Compute Cloud (EC2) which allows 750 hours per month, and Amazon Simple Storage Service (S3) which allows 5GB of standard storage.

The 'Billing Dashboard' (see Figure 5 above) is always easily available in the console banner and gives a breakdown of current expenditure balance per tool. Further in-depth analysis can be done at various granularity levels. Billing alerts can also be setup to send an email after certain usage thresholds. These alerts are not offered as default upon setup so it is important for all users, but particularly beginners, to be aware of their usage as costs can easily occur without noticing when tools such as EC2 are running.

To extrapolate for a few business user, the AWS online calculator [6] was used. For the use of a Windows OS with 8GB of Instance Memory and 10GB of storage per month, the estimated monthly costs are over €2,000. The advice to new users is to spend some time getting familiar with where the charges lie as they can be easy to miss sometimes (e.g. allowing an Instance to continue running, instead of stopping it once finished with it).

Configuration

Microsoft Azure (4.5/5)

User can change the web.config file easily directly from the Azure Portal. It is very useful in live site scenarios when something needs to be turned off or change. This avoids deployment process which is normally time consuming. User can enter key value pairs for such settings.

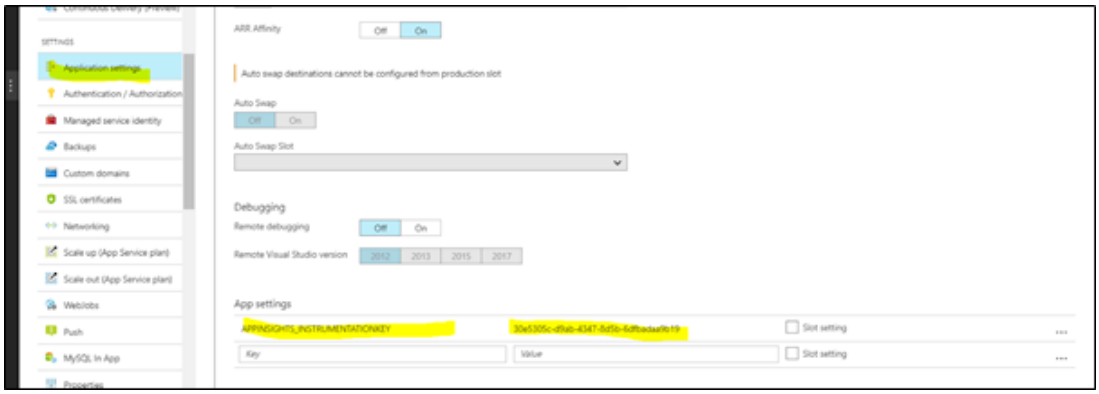

Figure 11 - Red Hat OpenShift

Other configuration options are auto scaling, custom domains, ssl certificates, authentication, deployment options, deployment slots and etc.

For storage we can configure following things from the portal: Replication, Access Tier, Secure Transfer, Performance, custom domains and etc.

Openshift (3/5)

We can change the key-value using Environment Form to add key-value pairs from config map or secret as environment variables.

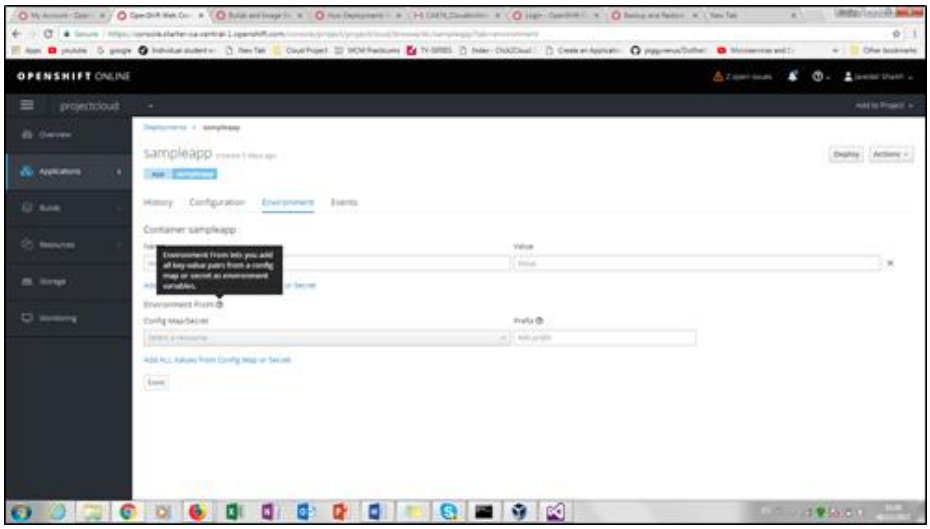

Figure 12 - Red Hat OpenShift

AWS (5/5)

Access to Configuration changes in AWS is quite simple, via the Elastic Beanstalk dashboard. Scaling, notifications, instances, etc are all easily accessible from here - see screenshot below. Once any changes are applied, the app will take a few moments to update, though usually less than 2 minutes.

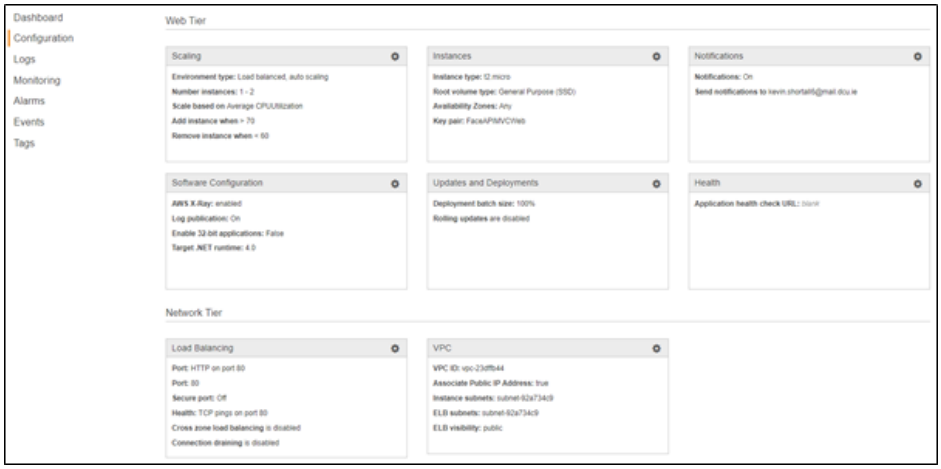

Figure 13 - AWS Console Configuration GUI

Backup resources and options

Microsoft Azure (4.5/5)

For website we are creating backup every two days (configuration in portal), and the website is stored in Blob storage as the backup image.

The Blob storage is read only access geo redundant storage i.e. it created a backup storage as 'secondary' The primary endpoint is https://faceapiweustorage.blob.core.windows.net/ and secondary endpoint is https://faceapiweustorage-secondary.blob.core.windows.net/

Screenshot for Backup of website:

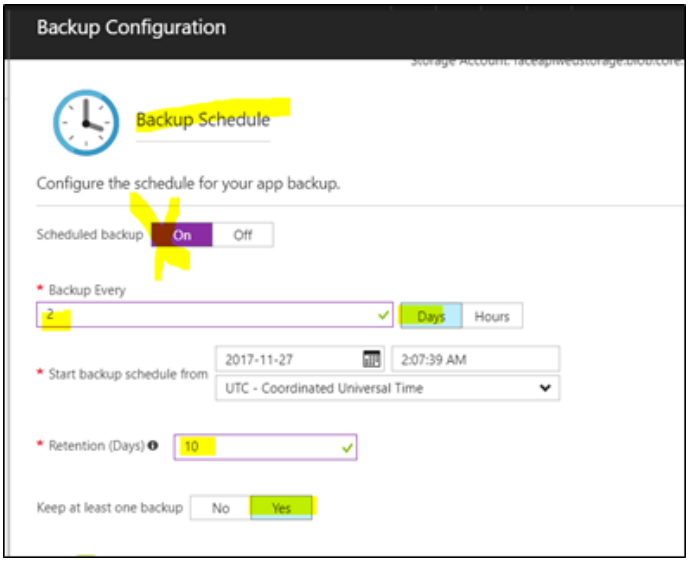

Figure 14 - Azure Website Backup

Red Hat OpenShift (2.5/5)

Has option to create backup at project level using Command Line Interface. To backup all the project with the exception of cluster objects like namespaces and object.
$ oc export all -o ymail > project.yaml
$oc rsync - to backup application data

AWS (5/5)

Offers two main Backup options: Images and Snapshots. Both are easily accessible from the EC2 Instances section.

Images contain the information necessary to launch an instance, which includes information regarding the root volume of the instance, which AWS accounts have access permissions, and mapping to know what the Instance is connected to.

A snapshot saves your data to S3 and records only the blocks which have changed since your last Snapshot. These are knowns as incremental backups and offers more efficient storage usage.

Scalability

Azure (5/5)

We tested out the auto scale options for Face API Website. In Azure portal we can set up multiple auto scale rules which are automatically triggered either on metrics based or schedule. In metrics we can set up rule to fire when CPU, network, memory etc. exceeds or falls above/below a value. In Azure portal we can see the history of auto scale, detailed events and also configure **notifications to get emails** whenever the auto scale happens.

Screenshot:

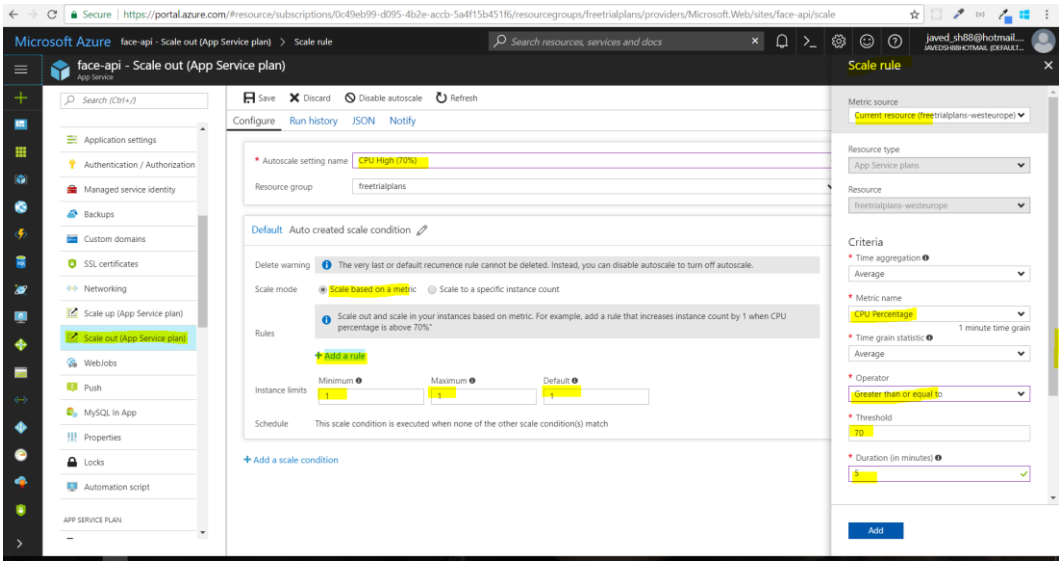

Figure 15 - Azure Autoscale settings

Red Hat OpenShift (3/5)

Auto scale parameter can be defined using project main menu from Application -> Deployment -> PROJECT NAME (sample) -> Auto scale AutoAlert Name = cpuhigh, Min Pods = 1, Max Prods = 2, CPU Request Target 70%. The CPU percentage request that each pod should ideally be using. Pods will be added or removed periodically when CPU usage exceeds or drops below target value - 70% - Defaults is 80%.

Openshift has a limited option supported by horizontal pod auto scalers

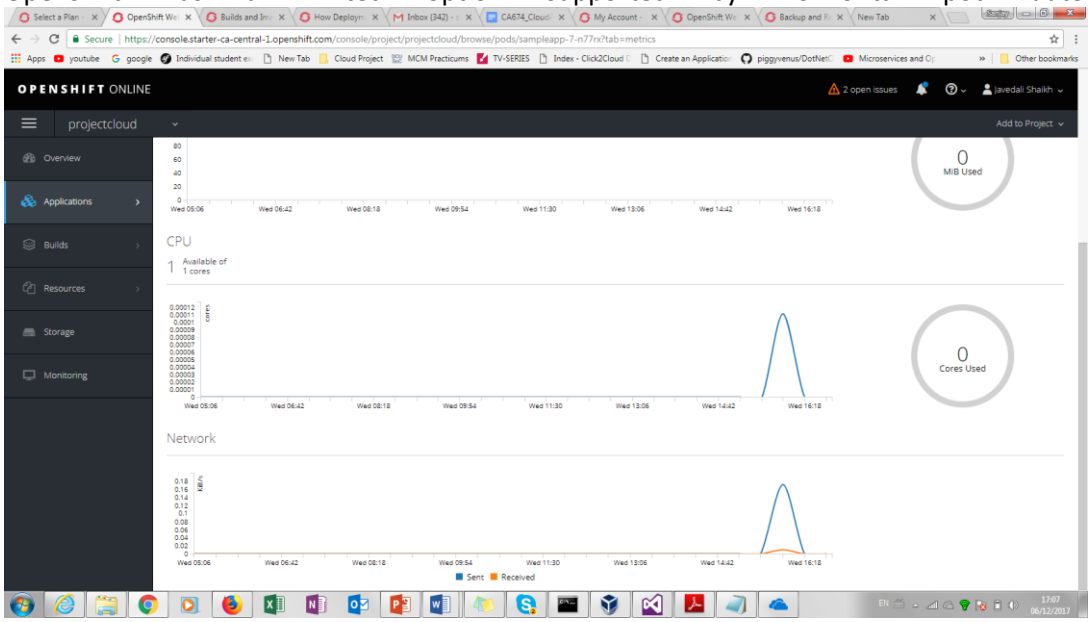

Figure 16 - Red Hat OpenShift

AWS (5/5)

The code forced the CPU load to raise to 80% for an extended period. With AWS, by setting Autoscale to trigger below 80% CPU load, we set Autoscaling into practice. The notification settings configured demonstrate that extra instances were added and removed as the autoscale progress, verifying that Autoscaling is functioning - see screenshot below. Autoscaling could also be configured to trigger for thresholds of latency, time, etc.

Figure 17 - AWS Autoscaling Instance Adding/Removing

Ease of Use/User-Friendliness

Microsoft Azure (4.5/5)

Does a fantastic job in providing one of the best UX. The portal is very smooth and UX friendly, plenty of documentation and large community.

Red Hat OpenShift (2/5)

We streamline application delivery easily manage application across the lifecycle and any of any environment. Openshift has some limitation on .net net application framework, there is no template available and documentation is very limited.

AWS (3/5)

Is the market leader and provides all cloud functionality that users and businesses require, navigation of the AWS console was not always a pleasant experience. This is partly down to certain areas of the interface feeling overcrowded with the array of options. The proliferation of AWS-jargon

is also off-putting - it can often feel that a beginner is required to have a reasonable knowledge of AWS and it's acronyms, even before learning about it!

Monitoring

Microsoft Azure (4/5)

Portal gives monitoring options for website and storage. For website we can see incoming requests, response time, CPU utilization, network and etc. Similarly for storage.

Figure 18 - Azure Monitoring Website

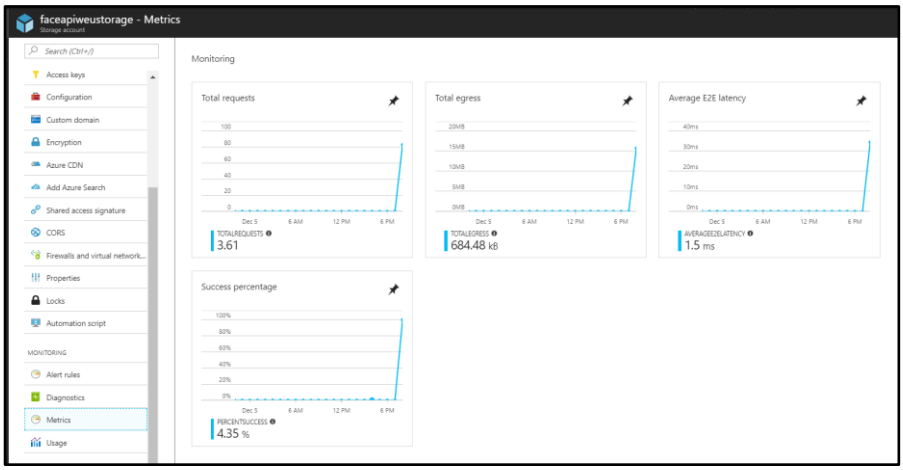

Figure 19 - Azure Monitoring Storage

Red Hat OpenShift (3/5)

Online provides monitoring features for each Pods and container level. We are monitor resources utilisation including Memory, CPU and Network.

Pod Level:

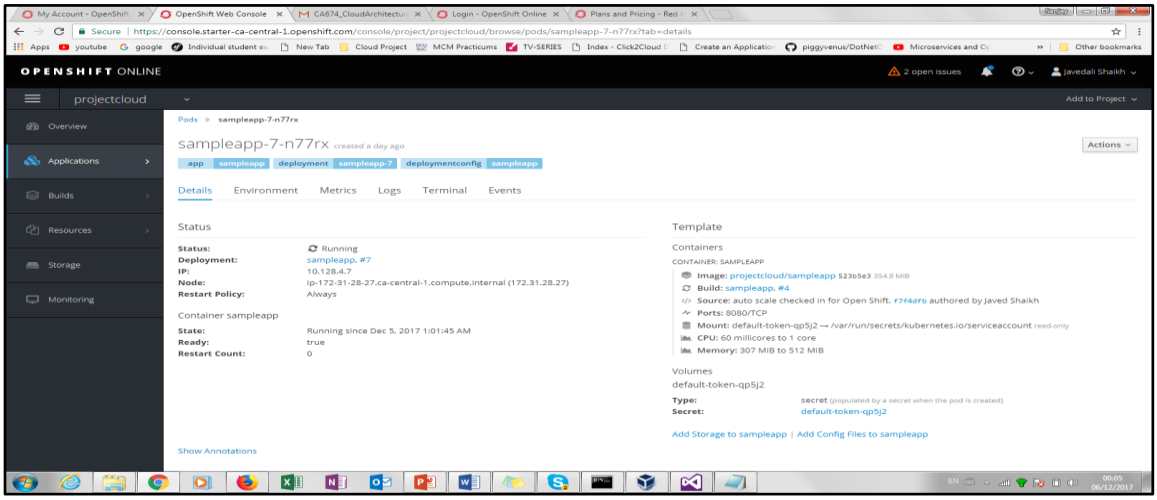

Figure 20 - Red Hat OpenShift

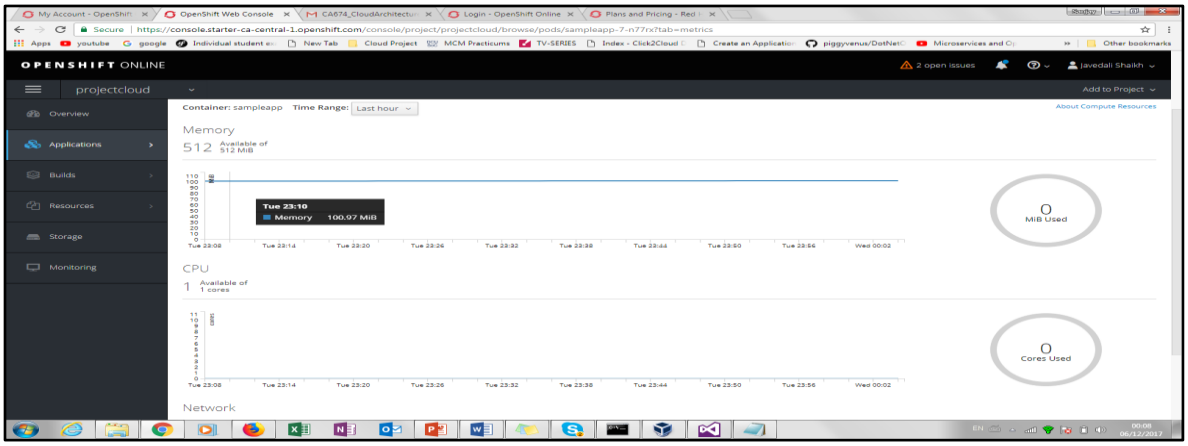

Figure 21 - Red Hat OpenShift

AWS (4.5/5)

Amazon Cloudwatch is the monitoring tool in AWS. It is relatively intuitive and easy to set up monitoring dashboards, alarms and events. There is an option to monitor your billing but is only available in the region of *US East (N. Virginia)*. If switching to that zone purely for monitoring of billing, note that it may impact the visibility you have of your Instances, Storage, etc.

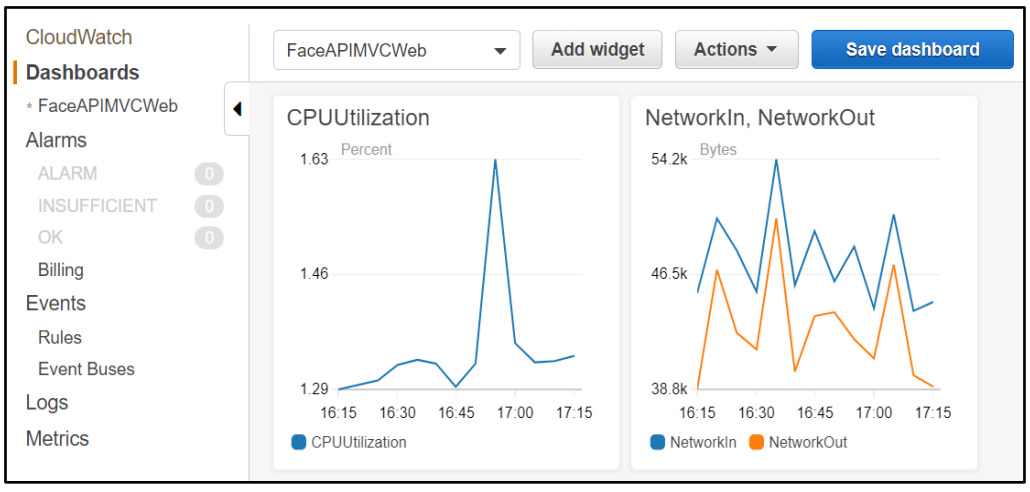

Figure 22 - AWS CloudWatch interface

Security

Microsoft Azure (N/A)

Provides various security options to all its services. The most popular is the Azure Active directory. This can be used in all sort of applications from server based website, to client based website, windows apps and etc. Also, other security options are username password, access keys, key value and etc.

Red Hat OpenShift (N/A)

Red Hat has a long history of managing the packages that make up Red Hat Enterprise Linux, including industry-leading responsiveness to security vulnerabilities and managing its online presence on Linux systems. Red Hat OpenShift Online is also proactively managed as part of the service.

Red Hat OpenShift runs on Amazon's EC2 cloud and inherits the security features of that platform. Learn more about the security of EC2 [8].

AWS (N/A)

Offers Security Groups at all levels. This can be in password form, or as an administrator you can allow access to certain users. A Security Group has to be assigned when launching an Instance, after which you can add/remove rules. Security was not a prominent feature of this project, therefore a rating is not applicable.

Challenges Faced

Azure

1. Blob storage was not getting created for almost half a day. Looks like there was temporary problems in creating blob storage under free trial. We were getting the exception 'Subscription cannot be found'. Whereas we were able to create blob storages under different subscriptions. This issue was almost present for half a day, which resulted in slight delay in our project
2. Though accessing the storages via REST API is pretty straightforward in Azure, but we faced some difficulties in the sample code we got from Azure. The sample code was adding some authorization header in HTTP request, which was not required, as we chose our storage to public. It took some time to figure out this. It would be great if the documentation and sample code are correct.

Red Hat OpenShift

1. Deployment of asp.net was not straightforward, and moreover Red Hat OpenShift does not support all version of as.net. To deploy specific asp.net
2. While trying to launch the sample website, getting environmental error - problem in loading page asp.net http://localhost:61878/ and IIS server. After analysis and google search got some understanding about the error and how to fix the problem. The problem was related to environment so after deleting file from the .vs folder and restarted Visual Studio https://developercommunity.visualstudio.com/content/problem/24939/iis-express-not-working.html
3. When deploying ASP .NET template sample page, below variable were missing so getting error. After adding environment variable from the error log in Environment tab, Rebuild the imager and it completed successfully.

Environment Variables added:

Name = DOTNET_STARTUP_PROJECT, Value = SampleWebapp/SampleWebapp.csproj

AWS

1. Deploying Applications: One obvious problem experienced was user permissions which caused a lot of problems in the early stages. The application url would open successfully and allow the selection of an image file, but would then hang indefinitely as it displayed “Uploading, please wait…..!” The solution to this took many attempts at different approaches and involved a number of steps. In EC2, the Instance’s inbound rules had to be edited to allow the Remote Desktop connection - shown below. The Remote Desktop could then be accessed by downloading and launching the Remote Desktop, as well as retrieving the Key Pair password. Note: If you cannot retrieve the Key Pair details, you may need to deploy the app from the beginning again and take special care with the Key Pair.
   Once in the Remote Desktop, the C:\inetpub\wwwroot folder was found and the security settings changed to allow the ‘USERS (EC2AMAZ-HCC12C0zusers)’ user group to ‘Modify’ - shown below. After these steps are completed, the app will function correctly.

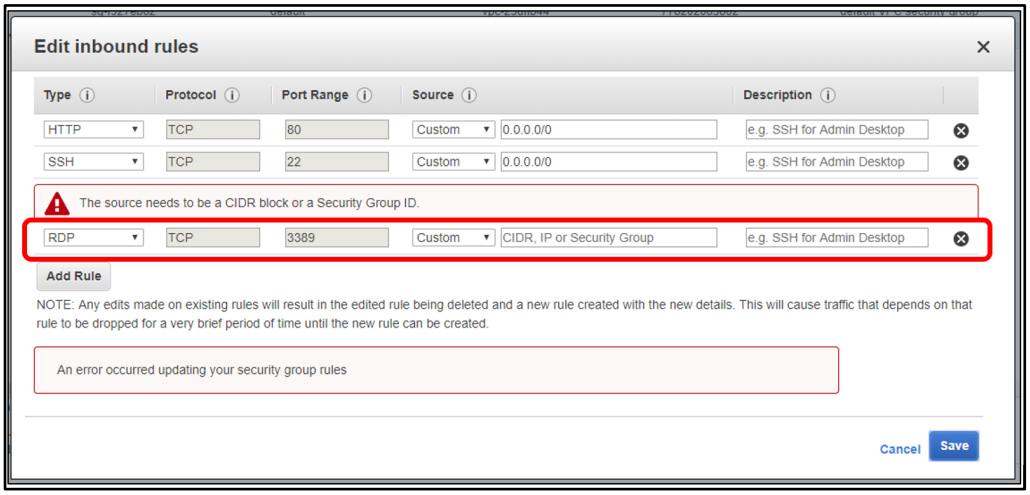

Figure 23 - AWS EC2 Inbound Rules interface

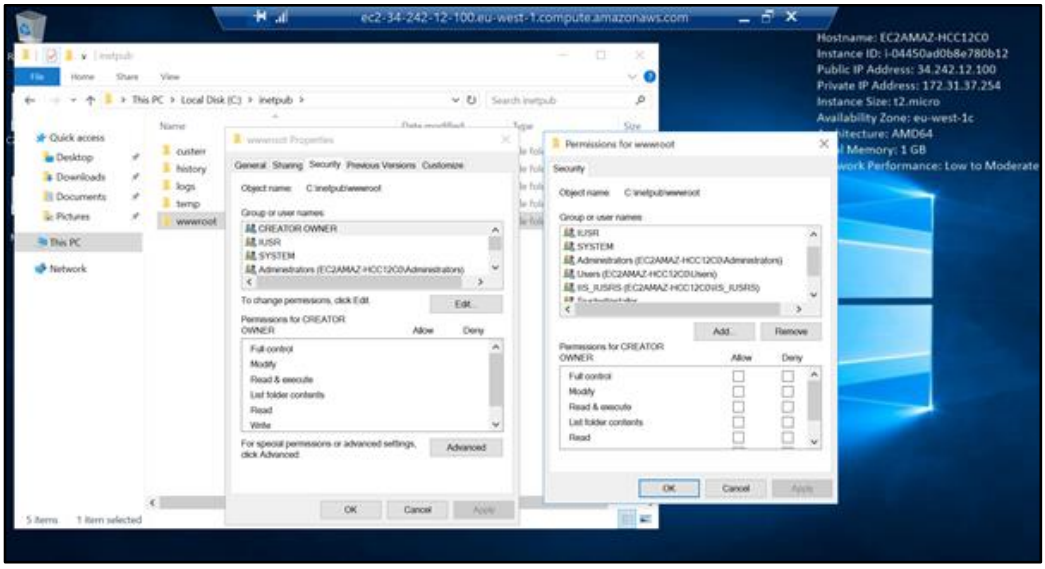

Figure 24 - AWS Remote Desktop

Conclusion

The final scores collated for each cloud platform is as follows:

- Microsoft Azure: 179.5
- Red Hat OpenShift: 110.5
- Amazon AWS: 169

The project has reached its objectives of demonstrating interoperability across cloud platforms as well SaaS through the use of the web app code. We migrated ASP.NET based website (first hosted on Azure) successfully to AWS with hardly any issues. Though, when migrating to Red Hat OpenShift, we face plenty challenges especially around support for particular .NET framework version. We experimented Red Hat OpenShift with latest .NET Core framework and worked out very well (with the help of SI2 templates). It would be good for developers if Red Hat OpenShift can add support to the most used .NET framework - ASP.NET MVC 4.5.

For Azure, we saw that it supports its own community (.NET) extremely well. There are multiple configuration, deployment options and is really developer friendly. Azure still has to do a little bit in the costing area and should come up with plans which suits even the start ups. For future, we will like to use Azure to host website based on completely different technologies like Linux, Python etc. and evaluate it against other cloud platforms.

For Red Hat OpenShift, we see it has great potential in the future, as it is adding new support for various technologies with its new version Red Hat OpenShift Online 3 and 3 Pro. Our recommendation for Red Hat OpenShift would be to give more resource in free plan, so that users can build good enough applications. For future, we would like to migrate the Face API website on .NET core and host on Red Hat OpenShift (we ran out of time to try this out).

While AWS is the market leader, it has reasonable challenges for a beginner due to certain interface features and an assumed level of understanding of AWS/Cloud terminology. Once familiar with the platform, these challenges significantly lessen and the products available cover a multitude of cloud and analytics functions.

References

[1]http://www.datacenterknowledge.com/business/microsofts-cloud-market-share-grew-more-anyone-elses-last-quarter-analysts

[2]https://www.business2community.com/cloud-computing/cloud-service-provider-comparison-will-next-big-provider-part-one-alibaba-01965914#vLEEF3SQY1hyfF2d.97

[3] https://developers.openshift.com/

[4] https://code.msdn.microsoft.com/Face-API-Using-ASPNet-MVC-40259a76

[5] https://azure.microsoft.com/en-us/support/forums/

[6] https://calculator.s3.amazonaws.com/index.html

[7] https://www.openshift.com/pricing/index.html

[8] https://www.openshift.com/policy/security.html