Master of Technology (Cloud Computing)

Continuous Assessment Instructions

Leveraging on Cloud Solutions

Table of Contents

Objective

The objective of this continuous assessment is for you to experience developing solutions on selected cloud computing platforms. Your team is free to choose any cloud platform – either those showcased in the classroom or the ones that your team has evaluated personally. You are required to perform the followings:

1. Identify and propose an appropriate *Cloud Use Case*.
2. Determine appropriate cloud eco system to implement this cloud use case.
3. Identify key characteristics as appropriate from the list provided.
4. Implement and demonstrate your proposal.

Administrative Details

Item	Requirement	Remarks
Team	4 students per team	Minimum 4 and Maximum 5 Members
Mark Allocation	40% for CA	60% from written exam
Project Proposal	***1st June 2017***	"Student Submission" work bin of IVLE. {*Lecturer is available for consulting on 27th and 29th May 2017*}
Implementation, Presentation & Final Report	***Full Time*** ***June 16th Friday 2017*** ***9:00 AM to 5:00PM*** ***Part Time*** ***July 1st Saturday 2017*** ***9:00 AM to 5:00PM***	Presentation to Lecturers Submit CD in specified format Time slots will be announced later

Proposal

The proposal should be captured in a 1~2 page document and submitted for approval. Due remarks would be provided by lecturer via email. In situations where similar project proposals are submitted, the first team that submits an acceptable proposal would be given the priority while the other teams would be asked to revisit the project proposals.

The proposal should be practical and solve a realistic problem that can benefit from cloud computing technologies. Credits are awarded for pragmatic and original work. The proposal should not be too simple and at the same time not be out of reach within the limited time frame. Please give due diligence to make sure that the nature of project is best solved using the service layer chosen.

Team are advised to self-enrol in IVLE groups and inform the lecturer if you have any issues. Kindly use IVLE discussion forum to find team mates if you can't group.

Criteria Listing

It is important to focus on the cloud computing aspects that to develop a traditional application from scratch. Teams are hence allowed to take any existing scenarios present a set of archetypal usage scenarios to illustrate a few of the ways in which cloud computing might be adopted and leveraged. Teams can pick any working traditional code base and launch it in the cloud.

Please note that each CA must worth at least a minimum of 6 points. The more points the team is able to accomplish, the higher the awarded scores. Also better quality of solution and execution would lead to more credits. Selected criteria listing is provided below:

Compute Platform	
Use of customized VM	1
Use of customised container/Docker/micro-services	1
Cloud-based Application Platform (e.g. Amazon Beanstalk, GAE, Azure Web App)	1
Cloud Native Application Platform (unique to cloud e.g.: Azure Logic App, Amazon Lambda, Kinesis, Azure Functions)	1
Database	
Creation or Migration to cloud based RDBMS	1
Creation or Migration to cloud based NoSQL	2
Use of caching strategies for transactions	1
Infrastructure	
Geographical redundancy (must demonstrate the failure case during demo)	2
Container dynamic auto scaling (auto scaling of containers based on the application load without needing to scale the number of VM)	2
Setting up of secure network zone (e.g DMZ + secure network zone)	2
Hybrid Cloud (on-premise + public cloud integration)	2
Installation and use of multi-node Open Stack container provisioning	2
Muti-tenancy using tools such as Mesos	2
API and other services	
Proper use of Analytics API/Services	2
Proper use of Machine Learning API/Services	2
Proper use of Common services (notification, email, sms)	1
Real-time/Stream analytics	2
IoT or ChatBot	2
Form Builder Services	2
Visualization Services	1
Social Collaboration Framework	1
Security and tenant credential management	1
Report Builder Services	2
Workflow engine or rule engine services	2
REST/SOAP based services	2

SaaS	
Innovative business model	1
Profitable pricing model (must simulate revenue and costing)	1
Design and implement a typical application as a Data-As-A-Service/ Business Process As A Service (BPaaS)	2
Design and implement application multi –tenancy and dashboard	2
Innovation	
Uniquely cloud (impossible or very hard to do using traditional technology on-premise). Cloud innovation relates to using cloud to renew or improve, and novelty is a consequence of the improvement.	2
Using cloud for access ubiquity and remote storage has other advantages. For instance, if you were to lose your mobile phone or your laptop, your applications and data would remain accessible, and all you would require is an ability to access the cloud environment.	2
Others	
Migration of existing application to cloud with significant modification (e.g. revamp from RDBMS to NoSQL)	2
Significant coding effort for creating an innovative application	2
Writing a step-by-step tutorial for part of what you. Tutorial should be either new (nobody has written such tutorial or better/unique compared to existing tutorial)	1

Cloud Case Studies for Inspiration

This section contains additional examples and analysis of cloud adoption case studies. Teams are free to derive inspiration from these case studies. We hope that these brief seed ideas that could hopefully trigger your innovation and imagination.

Links

Some other ideas which is inspired by this: https://developer.ibm.com/bluemix/2014/12/16/graduate-students-learn-innovate-on-bluemix/

Search Engine for Physical Notes (kind of Evernote clone)

Application that allows upload of images that could be a scan of your notes/bills/receipts, do an OCR in the background, feed the result into a search engine and provide API and web application to search and retrieve back the image.
(Possibilities - Compute (1), OCR Service (1), Search Engine (1), Object Storage(1), maybe some API gateway service (1), some coding (1-2))

ChatBot for CI

ChatBot is a machine-learning based conversational dialog engine which makes it possible to generate responses based on collections of known conversations. Connect Jenkins to a bot so that the bot will tell you if anything interesting happens with your monitored project. You

should be able to tell the bot about the projects that you want to monitor and what kind of events you are interested on
(Possibilities - Bot Framework (2), App Service (1), Jenkins on a VM (1), probably significant coding (1-2), perhaps a tutorial (1-2), maybe a NoSQL DB (2))

Emotion tracking

POC application that will capture the faces of people it detects, group the faces by similarity and then plot the emotions over a long period of time
(Possibilities - Face API (2), Emotion API (2), App Service to view the result (1), RDBMS (1), Object Storage (1), perhaps a tutorial (1-2), maybe significant coding (1-2))

Analytics on the Cloud

Normal enterprise analytics may not scale well for analytics needs of performance and elasticity. Often enterprises need to run regular analytics on customers, supply chains or manufacturing operations, say on a daily basis. Such jobs may run for a few hours on dedicated hardware, and occasionally require even larger capacity, thus leading to over provisioning of infrastructure. Using the cloud, the required infrastructure can be provisioned when needed and disbanded thereafter.

(Possibilities - MLib API (2), Streaming or Other Complex API (2), App Service to view the result (1), NoSQL (1), Object Storage (1), perhaps a tutorial (1-2), maybe significant coding (1-2))

Container Orchestration

Micro services the ability to create independent, decoupled, stateless RESTful services around a specific business capability interconnected via a reliable asynchronous fabric. With Micro services the ideal mechanism of being able to host these services is on light weight containers. Teams can look at taking a traditional application and migrating them to vendor specific container management and orchestration services.
(Possibilities App Service to orchestrate (2), RDBMS (1), Object Storage (1), State Management (1), Restful web service (1), significant coding (2))

Experiment with Spring Cloud

Spring Cloud is a toolbox for building cloud native applications using the Spring Java platform. It brings together a set of design patterns and use cases that will often be encountered in building such systems. This can be combined with the high productivity of Spring Boot, to provide out-of the-box production ready features including health checks, externalized configuration and many others.
(Possibilities – Spring Cloud (2) Spring Boot API (2), Data or Object Store API (2), RDBMS (1), perhaps a tutorial (1-2), maybe significant coding (1-2))

Proposal Submission

The initial project proposal should be uploaded in IVLE - Work bin - Student Submission Folder. Contents would include:

- Description of the project including its features that substantiates your use case.
- List of criteria that you plan to implement/experiment.
- Technologies and tools that will be used to develop this service.

- A diagram and a brief description of the high level architecture of your system including service installed nodes, relevant commercial/open source components used, coding work done and any API components.

PS: **Please do not exceed 3 pages.**

Project Demonstrations

Each team would be allotted 20 minutes to present and demonstrate the project. The presentation slides should be a good summary of the project report (maximum 5 slides). Each team should submit the following deliverables before the presentation. Major flows of the cloud use case implemented are to be demonstrated (the lecturers may suggest additional flows that are implied by the project proposal). It is recommended that all team members are present for the presentations.

Code Submission

The teams must submit the following finally after the demonstrations via IVLE:

1. A simple project report detailing the cloud use case, design and technical implementation details. The report should also elaborate on the innovative work done along with the lessons learnt. Please do not exceed 10 pages.
    a. The report must contain team member names, matric number, contributions and any other resource specific details.
    b. The report must be uploaded in the group's submission folder.
2. The relevant code and other installation/configuration details documents must be submitted. Related application source files, scripts, configuration files, etc. and related application packages must also be submitted.
3. A document containing the screen shots of working through the critical use cases must be submitted.
4. If virtual machines or Dockers are being used, configuration details of those virtual machines and copy of the working image file needs to be submitted via DVDs.