Cloud Architecture Design Task – Proposal

Background

There is a Computer Science (CS) department at a major university that accepts at least 200 new students per year. Each student is assigned a user login ID when they enter their program. Currently, the department is hosting their own servers on premise, but like that Cloud hosting might be the ticket!

This job contains the details of a written design proposal solution utilizing Cloud architecture that fits their requirements.

Requirements

Please refer to the following numbered list of requirements when they are referenced later:

1. Virtual Machines (VMs) will come in two flavours: Windows and Linux (Debian). If the latest versions are not available, then it should be discussed why the Cloud platform's VM offers are sufficient for the department's teaching mission.
2. Linux VMs should be able to deploy Docker containers, but should also have a selection of software that includes programming languages such as C, C++, Java, Python, Perl, Go, Scala, etc. as well as the RDMSs MySQL and Postgres. Faculty will wish to have various software packages installed and this should be easy for the system administrators to handle.
3. Windows VMs should be able to deploy Docker containers, but also support Visual Studio and the same set of languages and databases that Linux provides. Faculty will wish to have various software packages installed and this should be easy for the systems administrators to handle.
4. Each student should be able to authenticate with their student central login ID and password, and they should have their own Linux and Windows files.
5. Student files should be maintained for as long as the student is taking courses and for 1 year after their last enrollment. File storage should take advantage of the Cloud platform's different types of storage for lowest cast and greatest availability.
6. Obviously, everything should have a backup (particularly student accounts)
7. The load on the compute systems must be balanced so that the system expands to maintain appropriate response times when many students are using the system (eg. think the night before an assignment is due) while releasing unneeded resources to save money.
8. Monitoring of the system for load, security, and costs should be provided to the system admins.

Addressing Background Size

Considering that there are many students who take 5-year programs because of co-op, gap years, or stretching it out by taking less courses each semester, the minimum 200 new students each year will result in at least 1000 students in the CS department at a time. Providing a bit of leeway for extras and any students that may spend an extra year, as well as the few admin and manager accounts (including staff), 1250 is a good approximate number. Notably, this number will affect the number of VM instances necessary at a time, the storage size necessary, the number of credentials required (i.e. user login IDs), backup storage size, load balancing for the system, and system monitoring.

General Reasoning

Based on the needs and requirements of the CS department of a major university that were outlined in the background, the architectural solution should highly consider overall pricing, as well as the preferability of Canadian data centres and its impact on cost. In general, the design factors various criteria, such as location, cost, reliability, security, scalability, maintainability, convenience and ease of use for users and administrators. It's important to note that although cost is an important factor, universities (especially major ones) tend to have lots of funding and this design proposal prioritizes the other factors more than pricing to some extent. Factors such as reliability and scalability, as well as maintainability and ease of use were regarded as more important for its usage in a university learning environment. It's very important that the system is reliable and functional in order to support student, staff, and admins with using the system for activities such as teaching, development, and maintenance without unforeseen outages and things breaking down. When assessing a solution for this design proposal, the "Big 3" were examined mainly: Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP). This design proposal has opted to use AWS as the Cloud platform for the solution. It is one of the "Big 3" and has many benefits, including lots of documentation, high convenience, user-friendly aspects, reliable support, strong security, and easy-to-implement features. Factors that have gone into this decision will be explained in more detail with regards to requirements, but for now a general overview will be provided, as well as a depiction of the solution. Of course, pricing and location availability for Canadian data centres also played a part. One of the key strengths of AWS is the "breadth and depth of its services, with more than 175 across compute, storage, database, analytics, networking, mobile, developer tools, management tools, Internet of Things (IoT), security, and enterprise applications" (https://www.computerworld.com/article/3429365/aws-vs-azure-vs-google-whats-the-best-cloud-platform-for-enterprise.html). As a result, AWS often wins on developer functionality, as it is in this case as well, and has also done well to provide economic benefits to customers, so long as they properly grasp the pricing and metric of certain services and how these impact cost of the overall solution.

In order to make it easier to manage and maintain, a single Cloud platform will be used. This will minimize the learning curve for admins and new staff because AWS comes with more user-friendly features. In addition, expansion and modification also becomes easier since everything is streamlined for one Cloud platform, and AWS also supports easy integration of other Cloud services. Furthermore, implementation is much easier when everything is from the same Cloud provider, especially since AWS has lots of useful and helpful documentation. Moreover, monitoring becomes very easy to do using AWS as a single platform. Overall, effectively utilizing AWS as the only Cloud provider makes many things easier while also minimizing some risks of using a hybrid solution. Even though this may result in slightly higher pricing, AWS still tends to do well in comparison to costs of other Cloud platforms for storage that totals 10GB or less. The benefits of this proposal outweigh the slight increase in pricing for the

Cloud architecture solution, especially because costs are saved by the offset of less cost used on employees for training and managing the system.

Architecture Solution

The proposed Cloud architecture design includes the following AWS services as part of its solution: Amazon Elastic Compute Cloud (EC2) (including EC2 Autoscaling), Amazon Elastic Block Storage (EBS) (including Amazon EBS Snapshots), Elastic Load Balancing (EBL), Amazon Simple Storage Service (S3) (specifically Glacier), and Monitoring via Amazon Cloudwatch, optional Amazon GuardDuty, and AWS Cost Explorer. If there are any cases where the suggested design could be implemented with similar alternatives, this will be mentioned in the detailed requirement explanation.

Proposal

Overall, the solution would entail the university's CS department utilizing their own website of some sort that contains connection guides, FAQ, and help contact information (similar to the SOCS wiki for University of Guelph). Here, instructions of how a user can connect using their user login ID (eg. central login ID) and their default password (eg. student ID number). After a user knows this information, they can follow instructions to connect to either the of the two latest version VM flavours: Windows or Linux (Debian). After connecting to a VM via SSH using the appropriate instructions provided by the department, each user will be prompted to login, where they should use their central user login ID and password (which they can connect through a central login portal or via something like LDAP – whatever works best for them). Each of these VM flavours can deploy Docker containers and also has a selection of software that includes various required programming languages and more can be added by system administrators if desired. Please note that user accounts can be restricted access to many permissions such as installing and downloading more software, which can only be done through request of an admin and usually won't be necessary. However, it's important to note that one of the desired functionalities you request isn't beneficial to your solution: Visual Studio for Windows VM. This will be addressed later, as well as an alternative solution if you truly desire it after all.

With regards to storing all of the users' files and info, it will be done using two separate EBSs (one for Windows and another for Linux) since it was designed to be use for such purposes in relation to EC2 as the OS file systems are different. Notably, during autoscaling of EC2 VM instances, the same EBS can be used for multiple instances, and this won't be an issue since a student will be restricted to only one access at a time per OS. However, it's important to know that there is a limit to this scaling with an EBS. Furthermore, automated backups to S3 are provided via EBS snapshots, where Glacier of S3 for archiving backups is particularly useful, for both file data and user credential backup. In addition, EC2 has autoscaling options that can be used to help efficiently manage the number of VM instances required, while ELB can be used to help manage traffic and load to each EC2 instance. Lastly, monitoring is very straightforward and can be handled by Cloudwatch, optional GuardDuty, and Cost Explorer, as these even have built-in features for monitoring EC2 instances from the AWS.

<u>Virtual Machine Instances</u> (Requirements: 1, 2, 3)

AWS EC2 will be used to create and host VMs that will be used by the system solution and can support the latest versions of both Windows and Debian Linux (i.e. Windows Server 2019 and Debian 10.5). In addition, EC2 provides additional benefits with regards to its vast selection of images for many different OSs and applications, even allowing an option for custom images. This broad support will enable system admins to more easily create and facilitate deployments for multiple identical instances within the solution architecture. Furthermore, depending on load, more VM instances can be created or unused resources can be released. This will be elaborated on further when elaborating about load balancing and auto-scaling later.

Setup of a Linux VM can be completed via SSH connection, where deployment can even be automated through scripts (eg. Python Paramiko library, similar to A2). Given that the desired software and programming languages are supported by the OS, there should be no issues and things can be completed smoothly. In this case, it will be more beneficial to handle this part of the setup themselves, since paying additional costs for the Cloud provider to handle custom images and/or deployment is unnecessary. Additionally, Docker can be (installed if not already present and) used to deploy containers on VMs (similar to A2). This is also much more cost-effectively done by the department themselves, rather than paying additional costs to have it done for them. Notably, this process is similar for both Linux and Windows VMs; however, scripts used for Windows may require to be in a different language (eg. PowerShell).

Please note that the desire to use Visual Studio within the VM is strongly advised against. It will bring up the costs a lot because VM instances that support stronger, full GUI OSs will be required to use the Visual Studio IDE instead of just command line SSH VM instances. In terms of development, things can be done using terminal based editors (eg. vim) if someone needs to do something on the VM; however, it's standard practice to develop locally because it's more efficient and faster. Periodically, the code can be ported to the VM to be tested (eg. via GitHub repositories, etc.). In addition, management and users will need to have an additional software or service (eg. NoMachine) for connecting in order to utilize the full GUI and Visual Studio IDE instead of simply just an SSH command terminal. This comes with additional costs for both architecture itself, employee maintenance and management, as well as learning curve for both the staff and students. Also, this will affect performance of the system as well as the cost because running stronger GUI-capable VM instances will take more power. Although the slight changes necessary to include Visual Studio for Windows VMs have been outlined above, I advise against it because it is not beneficial to the architectural Cloud solution and brings more concerns that outweigh the advantages of doing so and it is recommended to use the aforementioned approach instead.

<u>User Login, Access, and Authentication</u> (Requirements 2, 3, 4)

Both admin and standard users will access the VMs (and their associated files) via SSH since both Windows and Linux VMs allow for the creation of users with username and password credentials. Using this aspect, each user will have credentials based on the university's central user login ID and password and it can be created by administrators using the school's central portal or credential system such as LDAP. For example, new users can be created using automated scripts (eg. bash) that take new students and staff from a list and create profiles for them using info from LDAP. Regardless of how this is done, it is something that the system administrators will be responsible for.

Each created user is given basic permissions that are set up by default. For instance, they can only access their own user folder which is where their files are stored. This is done atomically by both OSs and does not require extra work. If users desire more software or other features within the VM, it can be setup so that they are unable to do so themselves and must request such things from a system admin. For example, permissions would not allow standard users to install new software on the VM. This also addresses the ability for system admins to handle various software package installation. However, please note that by default, Windows Server 2019 does not seem to support SSH connections, so this must be set up using third part software first (eg. Open SSH). The following links will be helpful for user creation, password authentication, and setting up SSH: https://aws.amazon.com/premiumsupport/knowledge-center/new-user-accounts-linux-instance/, https://medium.com/@rakeshwrites/how-to-enable-password-authentication-in-aws-ec2-instances-26fbdddd74b0, and https://docs.microsoft.com/en-us/windows-server/administration/openssh/openssh_install_firstuse.

Storage and Backup Storage (Requirements 4, 5, 6)

AWS EBS will be used to store user files as the storage solution EBS was designed to be used with EC2 instances is very suitable for this cloud architecture solution. There will be two EBS volumes (one for Windows, one for Linux) because the two OSs have different file systems and each user's two "VM flavour" files will be stored separately. Notably, an EBS volume can be attached to multiple instances using the Multi-Attach feature in the case where load balancing and EC2 auto-scaling leads to the creation of more VM instances. Consequently, files can persist while instances are created and terminated from effects of EC2 autoscaling and load balancing. Although this may introduce concurrent access issues between multiple instances, if each user is restricted to their own user folder on the VM OS and is also restricted to one connection at a time, the issue is avoided. In addition, the EBS system allows admins to increase capacity as necessary without any downtime. For instance, capacity can be monitored, and extra storage can be provisioned to the resources or unused storage can be deallocated depending on load and traffic. However, it's important to note that there exists an alternative approach where each user (or a group of users) has their own EBS volume, but there is a limit to the number of volumes that can be attached to a single EC2 instance making this solution not scalable for hundreds of students. Please see the following links abouts EBS volumes and volume limits: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volumes-multi.html and https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/volume_limits.html.

Furthermore, EBS offers automatic snapshots via Amazon EBS Snapshots that can be used for storage and backups in addition to life cycle management regarding creation, deletion, and retention of snapshots in S3 (snapshots = $0.055/GB-month of data stored, restore = $0.83/DSU-hour/snapshot/AZ). In terms of backing up data and information, Amazon S3 Glacier will be used for the nature of infrequently accessed, archival data and the much cheaper price of storage ($0.0045/GB vs standard $0.023/GB). Since these backups will tend to be long-term as its required to hold user info files for at least a year after their last enrolment, this information won't need to be retrieved often. However, note that there are varying retrieval times from 1 minute ($0.033/GB) to 12 hours ($0.011/GB or bulk $0.00275/GB)), depending on the tier of request (varies from $11 to $0.055 or bulk $0.0275 per 1000 requests). However, note that the high tier requires a provisioned unit with separate additional cost ($110/provisioned capacity unit). Therefore, the decision between Glacier and standard S3 should be carefully considered due to the nature of emergency restore from Glacier backup requiring high tier, quick retrieval access. In most cases, management will know ahead of time and would be able to request returning users' information. In emergency cases where these backups are used to restore data (eg. EBS died and lost files), a high-tier request can be used to speedily retrieve info for fast recovery. By its nature, S3 replicates data for free across Availability Zones (AZs) (geographic locations), so if one AZ goes down (or the machines there are physically destroyed), data would still be accessible using other AZs that remain. As a result, it is a very durable and highly available service for data storage, which is important for backups as you can see for the previously mentioned scenarios.

<u>Load Balancing and Automatic Scaling</u> (Requirement 7)

Previously, EC2 Auto Scaling has been mentioned and is a feature available to use with EC2 at no extra cost. This refers to creating a group for automatic scaling, which can easily be done during setup (eg. from the AWS Console) and be configured for this solution based on different parameters, such as maximum spot price, spot allocation strategy, and on-demand strategy options. If the custom setup is applicable, it may be worth looking into Amazon EC2 Fleet to provision compute capacity and optimize, scale, performance, and cost. However, as this is most likely not necessary, we will leave this idea on the back burner and focus on EC2 Auto Scaling. Essentially, this automatically creates on-demand instances based on the monitored statistics (eg. from Cloudwatch). Basically, it helps to evenly distribute load and removes burden from the existing instances: if there is high traffic, then more are added, and for low traffic, it means that instances can be removed. These settings can be configured on multiple customizable settings. For instance, we can say that it will scale based on average CPU utilization where it will be flagged one it reaches a threshold of 70% or greater. Whereas ELB is used for balancing the load on the system, particularly regarding traffic to EC2 VM instances. For example, the Classic Load Balancing (CLB) version could be used for the EC2-Classic Network (multiple EC2 instances) and would operate at both the request and connection level, as well as support cross-AZ load balancing ($0.0275/CLB-hr, $0.0088/GB data processed). In general, ELB will help evenly distribute traffic and connections to the multiple EC2 instances (of that OS type: Windows or Linux). By utilizing both these features effectively, capacity is automatically and added and removed based on the utilization of the system, making it highly flexible, available, and elastic.

<u>System Monitoring: Load, Security, and Cost</u> (Requirement 8)

Monitoring of the system using AWS is very easy because of Amazon Cloudwatch and AWS Cost Explorer. Typically, Cloudwatch is present in the AWS Console and can be used to view the system from an overall perspective as well as per individual resources or services. For example, for EC2 Cloudwatch helps an admin monitor CPU, disk, and network usage (see picture below). Notably, each of the services used in this solution have support from Cloudwatch, which is a great benefit for monitoring. To continue, Amazon GuardDuty is a threat detection service that continuously monitors for malicious and unauthorized behaviours. Essentially, this will monitor the system and alert a system admin if anything is detected. It can also be easily utilized from the AWS console to monitor security without additional software to deploy or manage. Although it is possible to setup actions for GuardDuty that do more than just alerting a manager, it most likely won't be necessary for the use within a school learning environment. For example, it's not like there's super sensitive consumer info that's being protected and action should be immediately taken. Instead, it's more likely there is an error or something an admin should look at. However, it's important to note that this does accrue additional costs and is most likely not even necessary in this solution. The cloud architecture itself already has built-in security and permissions that should be sufficient enough for this solution, like using proper identity and access management when using the system (eg. IAM). Furthermore, Cost Explorer can be used to similarly visualize, understand, and manage costs and usage over time. For instance, Cost Explorer includes a default report that helps you visualize the costs and usage associated with the top 5 cost-accruing AWS services. In this case, this would be for the AWS account used for this cloud architecture solution. It provides a detailed breakdown, table view, report statistics, and even settings to look at specific hourly and resource level granularity. In addition, it has additionally support for managing costs, including a savings plan that can be utilized with EC2 usage. Please look at the images at the end of this section for some example visuals.

Cloudwatch can most likely be utilized within the free tier for the solution, but it's important to note the paid tier is pay-as-you-go and can be used if desired with some additional cost although not personally recommended in this design proposal. In my honest opinion, security used should be left up to the system admins since no requirements have been provided. At present, the default basic security is enough for basic common issues, and if something stronger is desired, there are many options available and I have mentioned two of them. For instance, GuardDuty, which continuously analyzes system flow, logs, requests, and responses, is charged at most $1.15/GB/month, being discounted at tiers due to large volumes, eventually reaching a rate of $0.17 for over 10TB in a month. To clarify, this is part of the design proposal although an alternative is to not remove it or replace it. An alternative to Cost Explorer would be AWS Cost & Usage Report, where both are around $0.01/request. Note that explorer hourly and resource level granularity will cost $0.01/1000 UsageRecords-month (defined as one line of usage – i.e. an EC2 instance running for 24hrs generates 24 distinct usage record at hourly granularity).

Expected Cost

	AWS	Azure	GCP
VMs	EC2	Virtual Machines	Compute Engine
Storage	EBS	Disk Storage	Persistent Disk
Backup	S3, EBS Snapshots	Blob Storage, Disk Storage snapshots	Cloud Storage
Load Balancing	ELB, EC2 Auto Scaling	Virtual Machine Scale Sets	Cloud Load Balancing
Monitoring	Cloudwatch, Cost Explorer, GuardDuty	Monitor, Cost Management, Advanced Threat Protection	Cloud Monitoring, Cloud Logging, Cost Management, Event Threat Detection
Estimated Cost Per Month	**$406.18/month (CAD)**	**$462.49/month (CAD)**	**$473.00/month (CAD)**
Estimated Cost Per Year	**$4874.16/month (CAD)**	**$5573.88/month (CAD)**	**$5676.00/month (CAD)**

Note: EBS Multi-Attach requires us-east AZ, but we know from this course working off us-east works fine. The benefit of the feature outweighs the small inconvenience of it not being Canada Central AZ. Also, AWS supports US more than Canada, which increases convenience and availability in a way. For consistency, the other estimations also use North Virginia (us-east) and destination as Canada Central if applicable. Also note, Canada Central source prices were more expensive anyways.

Above is a table that compares services from Azure and GCP for a similar architectural solution and will be used in calculations to compare estimated costs, which are in the last row of the table. It's important to note that these costs are based on cost calculators and are for the core components of the system: VMs and storage. In other words, aspect such as backup, scaling, load balancing, and monitoring are excluded from this calculation to keep the core comparison as accurate as possible since these additional services are to support the main components of VMs and storage. As such, the following characteristics were used for each platform's estimation in their respective cost calculator: pay-as-you-go pricing, 2 VMs (1 Windows, 1 Linux) each with 2 vCPUs and 8GB RAM (eg. *t2.large* for AWS meets this), 100% VM utilization/month (i.e. 24hrs/day, 7 days/week = ~730hrs by default), 1TB SSD storage per VM (2TB total), 1,000,000 storage transactions/month (if applicable). For other estimated costs, they can be extrapolated from the pieces in each of the previous paragraphs. For this table's purpose, we only looked at the core components regarding the cost estimation. Factoring in the other services, we would be looking at approximately $800 to $1000 per month at most. Typically, due to the nature of the load balancing and auto scaling, including the fact that it won't experience many users most of the time and usually have more "burst" activity of high traffic, we are looking closer to the lower end of $800/month overall. Potentially, we may even see as low as $600/month depending on usage, since that affects monitoring and cost estimations for storage, backups, snapshot, and data transfer.

<u>Summarized Advantages and Disadvantages to this Cloud Solution</u>

As you can see from the above table, the costs are relatively close; however, AWS (surprisingly) came out on top as the least expensive option, providing even more ground for why it was selected as the cloud platform for this design proposal. To summarize, this design proposes a full AWS Cloud architecture solution, which comes with the advantages of a relatively easy implementation (especially compared to previous on-site solutions) where no hardware (servers, etc.) is required physically, in turn removing the need to spend time and money figuring things out and trying to acquire, set up, and maintain on-site hardware for the system. It also allows for the use of high-quality infrastructure and services provided by AWS along with the potential for easy expansion and integration with more AWS services in the future. Overall, it's extremely reliable, easy to use, and highly customizable, which comes in especially useful when configuring settings to be optimized for this particular system, including cost settings. On the other hand, there are still a few disadvantages since no solution is 100% perfect and there will always be some trade-off. This way, there is almost no control over the specific hardware that is being used for the system even if you can specify some things. Moreover, there are still sometimes platform, region-wide (or AZ-wide) downtimes and outages that will be beyond you control. For example, AWS recently had an outage that would've affected many different services. In particular, this one affected some online textbook hosting and in turn affected many academic evaluations that were open book since the textbook was inaccessible during this period of time. Although rare, it is still important to know that these kinds of things can happen sometimes. Furthermore, as mentioned earlier, this solution may not be appropriate for storing sensitive and/or confidential information since the security levels are not up to par with the standards required at higher levels. In this case, central user login ID may not need to be stored directly in this system (eg. mentioned central portal, LDAP, that could be used instead) and might be avoided, but is still important to keep in mind. Lastly, as AWS is one of the "Big 3", the largest Cloud providers, this can make it a target for security threats as it would be a challenge for other to try and crack.

References

Entire AWS & AWS Documentation Websites:

“Amazon Web Services (AWS) - Cloud Computing Services,” *Amazon Web Services*. [Online]. Available: https://aws.amazon.com/.

AWS Documentation. [Online]. Available: https://docs.aws.amazon.com/.

Cost Calculators:

AWS Pricing Calculator. [Online]. Available: https://calculator.aws/. [Accessed: 16-Dec-2020].

“Google Cloud Pricing Calculator,” *Google Cloud Platform Pricing Calculator*. [Online]. Available: https://cloud.google.com/products/calculator.

“Pricing Calculator: Microsoft Azure,” *Pricing Calculator | Microsoft Azure*. [Online]. Available: https://azure.microsoft.com/en-us/pricing/calculator/.

Additional Resources:

maertendMSFT, “Installation of OpenSSH For Windows Server,” *Installation of OpenSSH For Windows Server | Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/windows-server/administration/openssh/openssh_install_firstuse.

R. Samal, “How to Enable Password Authentication in AWS ec2 Instances,” *Medium*, 04-Jun-2020. [Online]. Available: https://medium.com/@rakeshwrites/how-to-enable-password-authentication-in-aws-ec2-instances-26fbdddd74b0.

S. Carey, “AWS vs Azure vs Google Cloud: What's the best cloud platform for enterprise?,” *Computerworld*, 23-Jan-2020. [Online]. Available: https://www.computerworld.com/article/3429365/aws-vs-azure-vs-google-whats-the-best-cloud-platform-for-enterprise.html.