In this discussion, you will reflect on what you have learned and discuss architectural approaches for developing software for cloud platforms. As you formulate your response to this topic, think about the questions provided below to inform your post. Consider and provide a comprehensive response to the following questions:

* ~~What is important to know from a software development perspective about this topic and its relationship with traditional operating systems?~~
* ~~How does developing the software for each platform differ? How is it similar?~~
* ~~What does it mean for a cloud-based system to be serverless?~~ ~~How does this influence the software development process?~~
* ~~How does architecting applications for the cloud differ from traditional operating systems? Are there advantages to this method and, if so, what are they?~~
* ~~What are the cost differences between traditional and cloud-based operating platforms?~~
* ~~Do today's cloud-based platforms make traditional operating systems obsolete? Why or why not? Explain your reasoning.~~
* Which operating platform are you most likely to recommend and develop prototype code for? Why? Provide a rationale for your recommendation.

In your responses to peers, consider the following questions:

* Do you agree or disagree with your classmates' observations and recommendations as they compared and contrasted traditional operating systems with cloud-based operating platforms?
* What do you agree with and what might you add to your classmates' observations and recommendations? Perhaps there is something they overlooked?
* If you disagree with your classmates' observations or recommendations, what might you suggest that they do differently to ensure a more positive outcome?
* Did your classmates' posts answer the questions posed comprehensively? What might have been omitted from their response that leaves you questioning their observations or recommendations of one approach over the other?

Serverless Computing opens many gateways for software developers and operational functionality alike. Serverless does not mean the Server does not exist, it means the Server is hosted and maintained by a foreign company. This Service model is called “Infrastructure-as-a-Service”, or IaaS (IBM Cloud Education, 2019). Since the early 2010’s this business model has been gaining popularity, allowing the scaling of computing, network, and storage resources to match business workloads with a pay-as-you-go plan. This eliminates the need for owning and maintaining local servers, storage, and memory. This change in server design can in turn greatly empower the software development process. This empowerment comes from being able to focus more on the code and deployment than the server environment it’s being deployed from. Additional empowerment stems from being able to automatically scale CPU, RAM and Storage resources based upon need with the cloud provider. No more emergency hardware upgrades!

First and foremost, in the serverless environment security considerations change. Serverless functions are deployed in a Function-as-a-Service (FaaS) cloud platform, meaning that the function is executed in the Cloud hosts data center, network, servers and operating systems (How Secure Is Serverless Computing?, n.d.). As such, the security of the cloud platform is out of the Software Developers control. What is in their control is execution of code, logic, code, and the data utilized in that code. An additional factor is the modular nature of serverless applications, modular applications draw data from cloud storage, APIs, message queues and the Internet of Things (IoT, refers to various devices and all the data gathered on their sensors). Due to this highly variable and decentralized traffic encryption MUST be a top priority. Furthermore, considering this decentralized nature entails no storage of secrets as plaintext, using least priviledge principles, regularly checking dependencies of external libraries, and vigilant logging of server and function activity to ensure adequate data for debugging. Ultimately, these are all things that should be considered regardless of the development process. A locally owned and maintained server brings emphasis to the hardware and software considerations for the developer, whereas a cloud-based server brings with it more emphasis on the security of the data that’s being funneled through the cloud to fulfill business needs.

The differences in cost that a serverless environment brings are numerous. Factors to consider are physical servers, employee cost, security, and development time. To operate and maintain a physical server on-site in a business requires database engineers and architects, the physical server hardware, the configuration and licensing of server OS, and software developer communication with the database engineers and architects to ensure compatibility and security of the code being deployed. A cloud-based server eliminates the need for physical servers, reduces the number of employees needed to tend to the server-side piece of development, and eliminates the configuration and licensing of servers. Cloud Providers are very flexible in their offerings of virtual machines, ranging from General Purpose, Memory Optimized, and Compute Optimized (Suite, 2018). Furthermore, some of these providers can revolutionize how the developers in an organization communicate. A prime example of this is Databricks, a service that utilizes Amazon Warehouse Services to create a cloud objects storage of easily accessible and readily utilized data for an organization’s ETL, Machine learning and Analytics Workloads (Databricks, 2022). Databricks is a platform that houses all its database engineer, analyst and data scientist’s code is one GUI that more easily allows code documentation and sharing. Cloud-Based Computing is moving leaps and bounds towards eliminating traditional pain points in a business’s use of its data for data-informed decision making.

Today’s cloud-based operating platforms are in direct competition with traditional operating systems. Cloud-based operating platforms do not make Traditional Operating Systems obsolete, but they do drastically change how they’re used. One can argue that a traditional Operating System is still being utilized to build and create these cloud-based services. Cloud-based services are a new way of imagining and utilizing the hardware and Traditional OS to better serve business and consumer needs. Personal Data is the token of the future, with consumers providing data by utilizing the various devices hooked into the cloud-based infrastructure. This in turn provides the data to improve software and devices at a faster rate (Gizmodo, 2015). Traditional OS Development and Hosting is being phased out as the technological giants of the corporate world start to steer the utilization of technology in this decentralized direction.

If I had to recommend an Operating Platform to develop prototype code on, I’d recommend the Operating Platform a user is most comfortable utilizing. OS virtualization currently allows for any OS to mimic the environment of another OS. While there are many squabbles between developers about which platform is best, prototype code can be created on any OS for any environment a Software Developer can dream on. In the same way that an instrument is best realized by a competent musician, Software Development is best realized by a competent developer.

References

Databricks. (2022, July 26). *What is the Databricks Lakehouse? | Databricks on AWS*. Docs.databricks.com. https://docs.databricks.com/lakehouse/index.html

Gizmodo. (2015, June 4). *How Android Is Becoming the New Windows*. Gizmodo. https://gizmodo.com/how-android-is-becoming-the-new-windows-1708608151

*How secure is serverless computing?* (n.d.). ZDNet. Retrieved August 15, 2022, from https://www.zdnet.com/paid-content/article/how-secure-is-serverless-computing/

IBM Cloud Education. (2019, July 12). *what-is-Infrastructure-as-a-Service-IaaS*. Www.ibm.com. https://www.ibm.com/cloud/learn/iaas

Suite, Vm. (2018, August 27). *A Cloud Services Comparison of the Top Three IaaS Providers*. CloudHealth by VMware Suite. https://blogs.vmware.com/cloudhealth/cloud-services-comparison/