Activity 4: Deploy Application to Cloud and Research Part 2

Ryan Coon

CST-323

Professor Dan Neal

April 8, 2023

Source code URL:

https://github.com/rcoon1/cst-test-323

Deployment Screenshots:

AWS

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Google Cloud:

Graphical user interface, text, email

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, website

Description automatically generated

Cloud Computing Research:

**Read the assigned textbook required readings for this topic. Let's say you decide to leverage (i.e., consume) a REST API that is published from a social media website such as Facebook or Twitter. What are five relevant technical questions related to non-functional requirements and security that would you need to ask the vendor?**

Does this API conform completely to the constraints imposed by the REST standard?

Does this API offer integration with my service’s language/framework, or does it at least support a data format that I need for data parsing/manipulation (XML/JSON/etc.)?

Are my needs for consuming this API going to be met by the API’s usage limits and restrictions? If not, will I need to perform caching myself or restrict my own users’ activity limits?

Does the API offer authentication methods which allow users to give their explicit consent for actions taken on their behalf? (Think of a GitHub API which allows 3rd party applications to perform actions on user’s repositories for them).

Does this API have adequate developer documentation and support articles and communities where a team could learn from other customers’ use of the platform? Does the documentation set include a developer sandbox for getting better acquainted with its features?

**b. Compare the cloud features of Amazon AWS and Google Cloud. Present at least 10 features, explaining how they are similar and/or different. Explain your rationale.**

Regions/Availability: Both AWS and GCP offer various regions where users can host their solutions. AWS has over 40 Availability zones spread throughout 16 regions globally, and GCP has around 24 locations from which users can spin up new services.

Languages/Frameworks: Both companies offer PaaS solutions which cater to a smattering of more popular and common programming tools/languages. Lambda and GCP Functions support Node.js, Python, Go, and Java. However, AWS supports a handful of other languages too, including PowerShell, Ruby, and C#. The IaaS-level services can support whatever tools and languages developers can configure, of course.

 Database Offerings: Both providers offer hosting or nominal support for most popular database systems (Oracle SQL, Redis, MySQL, PostgreSQL). AWS and Google offer their own proprietary hosting solutions for long-term storage or archiving: Google BigQuery & AWS RedShift. AWS also offers RDS, which can host any of the main database engines along with their own proprietary relational database, Aurora.

Web Client/GUI: The AWS web client is certainly more mature than Google's cloud service. While GCP has a more modern-appearing, Material design interface, AWS seems to pack more configuration menus and options into their site overall.

Developer integration/SDKs/CLIs: Both companies sport a large number of SDKs to work with each of their more mature products. Both have official developer CLIs, which have the ability to manipulate most cloud resources just from the console. From personal experience, AWS appears to have a higher volume of developer documentation and tutorials for dealing with their code interfaces. GCP has a very streamlined documentation set, but it can leave developers searching elsewhere for more complex examples or explanations.

Pricing: This all depends on the needs of the individual customer, but overall these companies both offer comparable services for around the same cost since they are direct competitors. They both offer a free trial tier that gives roughly the same value (~$300 service credit) to new users.

Security: GCP's security interface is less straight-forward than the one AWS uses for managing its IAM permissions, in my opinion. AWS, as the more mature service, boasts a vast scope of granular security options. Its complexity can be a high barrier for new developers or teams to get used to. However, this makes AWS the go-to provider when security is a primary concern.

Scalability: Both companies offer instantly-scaling NoSQL databases and serverless computing. There are differences in setting configurations and limits for containerized services and their PaaS solutions, but there aren't any harsh disparities between what both can offer.

IaaS (Private Cloud): Both providers offer robust private cloud offerings. GCP has less, for sure, given that it's Compute Engine is the only product their offer that's purely IaaS. GCP largely got its start as a PaaS provider, whereas AWS began as an in-house IaaS provider for Amazon itself.

Support: AWS has the more mature support team and article set, but GCP also offers comparable tiers of near-instant support help, all the way up to getting aid with architecting services on their platform. AWS is likely slightly cheaper at scale, but both of them offer roughly the same features to each similar-tier support plan.

**c. Identify five technical or business limitations that could restrict an existing application from being deployed onto any of the cloud platforms that were utilized in Topics 3 or 4. Explain your rationale.**

Limitations restricting applications from AWS or GCP deployment include vendor lock-in, compliance and data privacy concerns, legacy systems and technologies, network latency and performance issues, and cost management challenges. Thorough assessment and planning can help address these limitations.

There are several technical and business limitations that can restrict an existing application from being deployed onto Amazon Web Services (AWS) or Google Cloud Platform (GCP). Here are five such limitations:

Vendor Lock-In: Migrating an existing application to a cloud service may lead to vendor lock-in, which means that the application becomes dependent on the proprietary tools, APIs, and services provided by the cloud vendor. This makes it difficult to switch providers or move the application back on-premises. It is essential to assess the extent of dependency on specific services and the migration costs involved.

Compliance and Data Privacy: In certain industries like healthcare, finance, and government, there are stringent regulations in place for data privacy and security, such as HIPAA, GDPR, and FedRAMP. While AWS and GCP offer various compliance certifications, it's important to note that not all regulations may be covered. Before migrating, it's essential for businesses to confirm that their selected cloud provider satisfies their specific compliance requirements.

Legacy Systems and Technologies: Some existing applications rely on legacy systems or technologies that are not supported or are incompatible with cloud infrastructure. For example, certain applications may require hardware components or specific operating systems that are not available on AWS or GCP. It is essential to identify such dependencies and assess if they can be replaced or if the application can be re-architected.

Network Latency and Performance: Migrating applications to the cloud may introduce network latency and affect performance, especially if the application has real-time or low-latency requirements. This is particularly relevant for applications that require data to be transferred between on-premises systems and the cloud or those that rely on low-latency communication with end-users. It is crucial to evaluate the potential impact on performance and consider solutions such as edge computing or hybrid cloud architectures.

Cost Management: While cloud services can offer cost savings through pay-as-you-go pricing and resource scaling, they can also lead to unexpected costs if not managed properly. Businesses must consider factors such as data transfer costs, storage, compute resources, and licensing fees. It is essential to conduct a thorough cost analysis and establish proper cost management practices to avoid unexpected expenses.

References:

AWS Well-Architected - Build secure, efficient cloud applications. (n.d.). Amazon Web Services, Inc. https://aws.amazon.com/architecture/well-architected/

Cloud Compliance & Regulations Resources | Google Cloud. (n.d.). Google Cloud. https://cloud.google.com/security/compliance

Compliance Programs - Amazon Web Services (AWS). (n.d.). Amazon Web Services, Inc. <https://aws.amazon.com/compliance/programs/>

Free. Amazon. Retrieved April 4, 2023, from https://aws.amazon.com/free/?all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free+Tier+Types=%2Aall&awsf.Free+Tier+Categories=%2Aall

Google. (n.d.). Cloud computing, hosting services, and apis | google cloud. Google. Retrieved April 4, 2023, from https://cloud.google.com/gcp?utm\_source=google&utm\_medium=cpc&utm\_campaign=na-US-all-en-dr-bkws-all-all-trial-e-dr-1605212&utm\_content=text-ad-none-any-DEV\_c-CRE\_491349594127-ADGP\_Desk%2B%7C%2BBKWS%2B-%2BEXA%2B%7C%2BTxt%2B\_%2BGoogle%2BCloud%2BPlatform%2BCore-KWID\_43700064423315751-aud-388092988201%3Akwd-26415313501&utm\_term=KW\_google%2Bcloud%2Bplatform-ST\_google%2Bcloud%2Bplatform&gclid=Cj0KCQjwla-hBhD7ARIsAM9tQKvTESTe5d4YR5r9PuBB-i4iG\_xaD7vEyN0aumlvZ2FqfQZNF4VOb5oaAgb5EALw\_wcB&gclsrc=aw.ds&hl=en

Google Cloud Architecture Framework. (2022, July 12). Google Cloud. https://cloud.google.com/architecture/framework

Kavis, M. J. (2014). Architecting the cloud: Design decisions for cloud computing service models (SaaS, PaaS, and IaaS). Wiley.