Activity 3: Deploy Application to Cloud and Cloud Research Part 1

Ryan Coon

CST-323

Professor Dan Neal

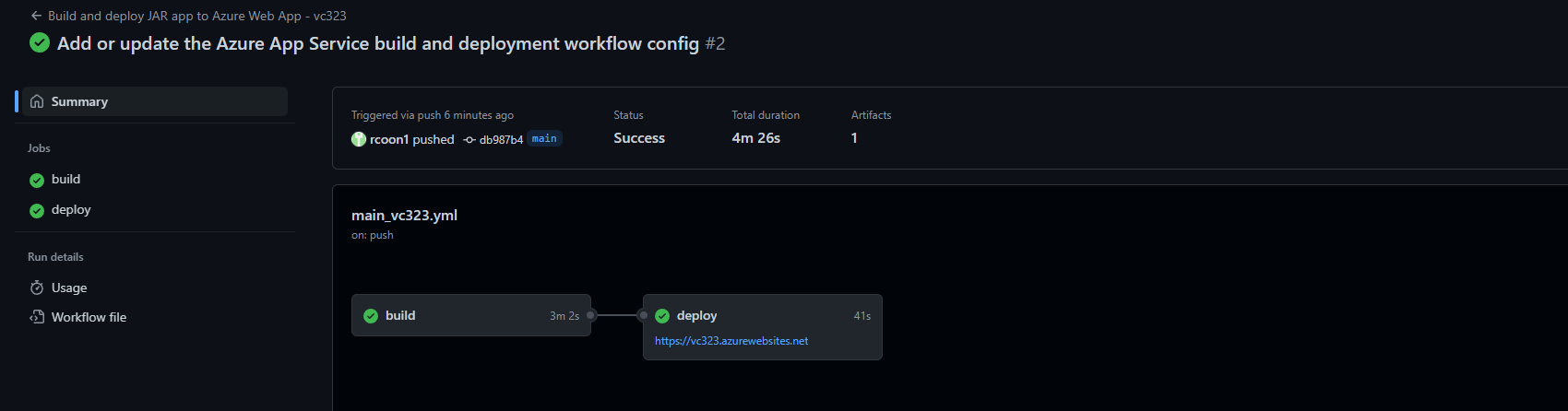
March 26, 2023

Final database and application design:

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

Contains source code and sql database information.

Microsoft Azure Deployment:



Graphical user interface, text, application, email

Description automatically generated

1. Step-by-step instructions used to configure and deploy the test application.

Click create a webapp. Name it. Then click deployment and start deployment. Choose CI/CD and connect to github. Choose the account and branch. It will put a run file in there for github to compile on deployment. Go over to github and click actions. You will then click deploy and it will run the deployment from there.

b. Challenges encountered during the database and application deployment.

Was finally able to get it to deploy, but the webpage times out every time I go to open it.

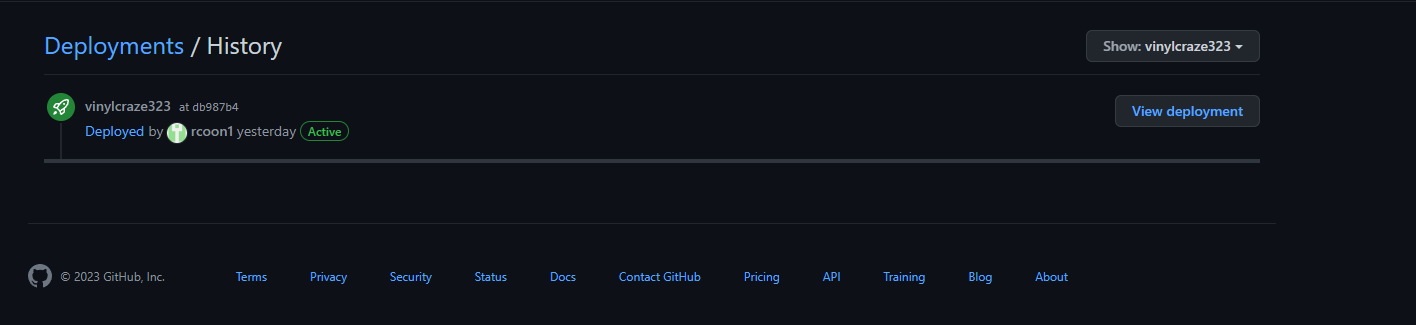
1. A screencast of the test application being run on the cloud platform.

https://www.loom.com/share/f857726c9e184deb8492592381fcf35e

Heroku Deployment:

Graphical user interface, text, application, email

Description automatically generated



1. Step-by-step instructions used to configure and deploy the test application.

Went to the Heroku website and created an account. I had to purchase the eco plan to be able to add an application. Once this was done I created a new app on Heroku. Then went to the deployment section to connect my github to it. Chose the file and branch to deploy then clicked deploy. Heroku took care of the rest.

b. Challenges encountered during the database and application deployment.

Deploying with Heroku was much more simple than with azure. I found a free database mysql host that I would use as my database. All I had to do was connect my github to Heroku, choose the file and branch and click deploy. It took care of everything. So easy.

1. A screencast of the test application being run on the cloud platform.

https://www.loom.com/share/f857726c9e184deb8492592381fcf35e

Cloud Computing Research:

**Read the assigned textbook required readings for this topic. Identify eight "worst practices" when migrating your application to the cloud. Identify the worst practice in two to three sentences, then provide three to five sentences on how to prevent the worst practice from becoming a risk or issue during an application cloud migration.**

1. Not understanding the differences between cloud computing models:

Cloud computing models, such as IaaS, PaaS, and SaaS, offer different capabilities, benefits, and costs. It is important for organizations to understand the distinctions when selecting the right model for their application. To prevent this worst practice from becoming a risk, organizations should thoroughly research and understand the differences between the models, consult with a cloud expert, and create a plan outlining the model that will best support the application's needs.

2. Not performing a comprehensive cost analysis:

 Organizations should consider upfront and ongoing costs when migrating to the cloud. To prevent this worst practice, organizations should perform a comprehensive cost analysis to determine the best service provider, model, and pricing structure for their application. Organizations should also create a budget and track costs to ensure they remain in line with their expectations.

3. Not performing a comprehensive risk analysis:

Before migrating to the cloud, organizations should perform a comprehensive risk assessment that takes into account factors such as the security of the cloud provider, compliance requirements, and data privacy. Organizations should create a risk assessment plan that outlines the risks associated with the move and how they will be addressed.

4. Not performing proper testing:

Before migrating an application to the cloud, organizations must perform testing to ensure the application functions as expected in the new environment. To prevent this worst practice, organizations should create a testing plan that outlines the scenarios and environments that must be tested. Organizations should also create a backup plan in case of any issues that arise during testing.

5. Not having a plan for monitoring and scaling:

Organizations must have a plan for monitoring and scaling their cloud-based application. To prevent this worst practice, organizations should create a monitoring plan that outlines the metrics that should be tracked and the tools that should be used. Organizations should also create a scaling plan that includes the criteria and processes to be used when scaling the application.

6. Not having a plan for data migration:

Organizations must have a plan for securely migrating data to the cloud. To prevent this worst practice, organizations should create a data migration plan that outlines the timelines, processes, and security measures to be used. Organizations should also create a backup plan to ensure that data is properly migrated and protected.

7. Not having a plan for security and compliance:

 Organizations must have a plan to ensure their application is secure and compliant with applicable regulations. To prevent this worst practice, organizations should create a security and compliance plan that outlines the processes and tools that should be used to protect the application. Organizations should also create a plan for monitoring and verifying the security and compliance of their application.

8. Not having a plan for rollback:

 Organizations must have a plan for rolling back to a previous version in case of any issues. To prevent this worst practice, organizations should create a rollback plan that outlines the scenarios and processes to be used in case of any issues. Organizations should also create a backup plan to ensure that the data is properly backed up and protected in case of any issues.

b. Compare the cloud features of Microsoft Azure and Heroku. Present at least 10 features, explaining how they are similar and/or different. Explain your rationale.

 10 features that are different between these two:

Functionality: Heroku is a platform as a service (PaaS) that serves the singular objective of making it simple for users to create and deploy web apps in the cloud. On the other hand, Microsoft Azure is a complete public cloud service featuring PaaS, IaaS, and SaaS features. Users of Azure can benefit from the full Microsoft ecosystem by utilizing various integrations (such as the Microsoft SQL Server database).

Pricing: Monthly fees for Heroku dynos range from $25 to $50 for subscribers to the "production" tier. Azure price is too complicated to summarize here due to the huge range of possibilities, but the platform offers both free and pay-as-you-go tiers depending on consumption.

Usability: Heroku is easier to use than other public cloud PaaS providers like Azure, AWS, and GCP because to its single-minded concentration. While creating and deploying web apps, Heroku greatly reduces the amount of time and effort required by users.

Customer support: Heroku has "standard" and "premium" support packages, with the latter promising response times of one business day and 24/7 assistance. Azure provides many levels of customer service, including premium help for business users.

Scalability: Both platforms are scalable, although Azure offers more flexible scaling options and can expand to a higher capacity than Heroku.

Deployment: Heroku provides an intuitive deployment approach that allows you to just push your code to the platform, and it takes care of the rest. Although setting up Azure can be more difficult, it offers greater customization choices.

Support: Whereas Heroku only offers email help, Azure offers a variety of support options, including 24/7 phone and email assistance.

Data management: Azure offers a wider selection of options for managing and storing data, including data warehouses, NoSQL databases, and SQL databases. PostgreSQL databases are primarily used by Heroku.

Security: Both platforms have strong security features, such as access control, firewalls, and encryption. For enterprise-level apps, Azure offers more sophisticated security choices.

User Interface: Compared to Heroku, Azure provides a more straightforward user experience with fewer options and functionality.

  10 similarities between these two:

Microsoft Azure and Heroku are both cloud-based platforms, which implies that they both provide hosting and computing services online.

Application Deployment: Both platforms allow for the deployment of web applications and give developers the tools they need to do so quickly and easily.

DevOps integration is a feature that both platforms offer, giving programmers the ability to automate the development, deployment, and testing processes.

Scalability: Users can scale their applications up or down in accordance with changing requirements thanks to scalability features offered by both platforms.

Continuous Integration and Continuous Deployment (CI/CD): Continuous Integration and Continuous Deployment (CI/CD) workflows let developers automate the process of creating, testing, and delivering code changes. Both Azure and Heroku offer CI/CD workflows.

Open Source: Both platforms enable developers to use well-liked open-source frameworks and languages since they support open-source technologies and tools.

Both platforms offer containerization technologies, like as Docker, which make it simple to bundle and distribute applications across many environments.

The ability to deploy apps across several clouds is provided by multi-cloud deployment, which is supported by both Azure and Heroku.

Monitoring and analytics: Both platforms provide capabilities for monitoring and analytics, allowing users to keep tabs on their applications' performance and usage.

Integrations with third-party tools and services are provided by both platforms, enabling users to expand the functionality of their applications.