Build Cloud Test Application & Research

Kaya Nelson

Grand Canyon University: CST-323-O500

Nov 11, 2024

**Deploy Application**

**GitHub URL: https://github.com/Kdeshun/CST-323-activitys**

This activity involves deploying the cloud test application onto both Microsoft Azure and Heroku while conducting research on cloud deployment best practices and comparing the features of these platforms. In this activity, students begin by creating accounts on both cloud platforms. They then provision the necessary containers and services required to support the application and its database, which includes configuring the database using the appropriate DDL scripts. Following this, students deploy their application code and thoroughly test both the application and database to ensure functionality.

The process culminates in a screencast demonstration showcasing the operational capability of the test application on the cloud platforms. Additionally, students engage in research to identify and document eight "worst practices" related to cloud migration, along with strategies to mitigate these risks. They also compare at least ten features of Microsoft Azure and Heroku, explaining similarities and differences in their offerings. This activity not only enhances practical deployment skills but also deepens understanding of cloud infrastructure and best practices.

**ER Diagram**

As of this milestone, the test application requires only one database table to function.

A black background with white text

Description automatically generated

**UI Designs**

A screenshot of a computer

Description automatically generated

The UI design should prioritize user experience, ensuring that users can easily navigate through the application, create, view, and manage their posts without confusion. Utilizing frameworks like Bootstrap can help in implementing a responsive and visually appealing design.

**Deploy Application to cloud & research**

**Weighing the Models: Comparison of Application Deployment Models**

When it comes to deploying applications in the cloud, organizations have several models to choose from, each with its own strengths and weaknesses. The primary deployment models include Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Below is a detailed comparison of these models, highlighting their key characteristics, advantages, and disadvantages.

**1. Infrastructure as a Service (IaaS)**

**Overview:**  
IaaS provides virtualized computing resources over the internet. Users have control over the operating systems, storage, and deployed applications while the provider manages the infrastructure.

**Advantages:**

* **Flexibility and Control:** Users have full control over the infrastructure, enabling them to customize configurations to meet specific needs.
* **Scalability:** Resources can be scaled up or down based on demand, making it suitable for variable workloads.
* **Cost Efficiency:** Pay-as-you-go pricing allows users to pay only for the resources they use, reducing unnecessary costs.

**Disadvantages:**

* **Management Overhead:** Users are responsible for managing the infrastructure, including security and maintenance, which can be resource-intensive.
* **Complexity:** Requires technical expertise to set up and manage infrastructure effectively.

**2. Platform as a Service (PaaS)**

**Overview:**  
PaaS provides a platform that includes hardware and software tools over the internet, allowing developers to build, deploy, and manage applications without worrying about the underlying infrastructure.

**Advantages:**

* **Development Efficiency:** PaaS offers integrated development tools, reducing the time needed to build applications.
* **Focus on Development:** Developers can concentrate on writing code and developing features rather than managing infrastructure.
* **Automatic Scaling:** Many PaaS solutions include automatic scaling features, adjusting resources based on application demand.

**Disadvantages:**

* **Vendor Lock-In:** Applications can become dependent on the specific tools and services of the PaaS provider, making migration to another platform challenging.
* **Limited Control:** Users have less control over the underlying infrastructure, which may restrict customization options.

**3. Software as a Service (SaaS)**

**Overview:**  
SaaS delivers software applications over the internet on a subscription basis. Users access the software through a web browser without needing to install or maintain it locally.

**Advantages:**

* **Easy to Use:** SaaS applications are typically user-friendly and require minimal setup, allowing for quick adoption.
* **Cost-Effective:** Subscription pricing reduces the need for large upfront investments and ongoing maintenance costs.
* **Accessibility:** Users can access applications from any device with an internet connection, enhancing flexibility.

**Disadvantages:**

* **Less Customization:** SaaS applications often offer limited customization options, which may not meet specific business needs.
* **Data Security Concerns:** Storing sensitive data on third-party servers raises security and compliance issues that organizations must address.

A table with text on it

Description automatically generated

When comparing different application deployment models, it is essential to consider various factors such as control, management, scalability, cost structure, customization, ease of use, and specific use cases. The three primary models—Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS)—each offer distinct advantages and disadvantages that can significantly impact an organization's decision-making process.

IaaS provides users with high control over the infrastructure, allowing for extensive customization and scalability. However, it requires significant management effort and technical expertise, making it more complex to set up and maintain [[3]](https://spacelift.io/blog/cloud-deployment-models). PaaS, on the other hand, streamlines the development process by offering integrated tools and automatic scaling, which can enhance efficiency. Yet, it may lead to vendor lock-in and offers less control over the underlying infrastructure [[3]](https://spacelift.io/blog/cloud-deployment-models). SaaS applications are user-friendly and accessible from any device, making them easy to adopt. However, they often come with limited customization options and potential data security concerns due to reliance on third-party servers [[3]](https://spacelift.io/blog/cloud-deployment-models).

In summary, the choice of deployment model should align with an organization's specific needs, technical capabilities, and strategic goals. Understanding the strengths and weaknesses of each model can help organizations make informed decisions that optimize their application deployment strategies.

**Reference:**

1. [A reference model for deploying applications in virtualized environments - PMC](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8073171/)
2. [Modeling Different Deployment Variants of a Composite Application in a Single Declarative Deployment Model](https://www.mdpi.com/1999-4893/15/10/382)
3. [Cloud Deployment Models - Types, Comparison & Examples](https://spacelift.io/blog/cloud-deployment-models)