Khumo-Thato Chabeli

st10448834

CLDV6211 Cloud Development A POE Part 1

Contents

[Links 3](#_Toc194933591)

[URL of the deployed Web App 3](#_Toc194933592)

[GitHub 3](#_Toc194933593)

[Sections for Solutions to Theoretical Questions: 3](#_Toc194933594)

[Cloud Computing Basics 3](#_Toc194933595)

[ERD Diagram 6](#_Toc194933596)

[6](#_Toc194933597)

[Screenshots of the web application 7](#_Toc194933598)

[Reference List 9](#_Toc194933599)

# Links

## URL of the deployed Web App

eventeasebookingapp-guehgbcgf9h8h0fp.southafricanorth-01.azurewebsites.net

## GitHub

https://github.com/KChabeli/ST10448834\_CLDV6211\_Part1.git

# Sections for Solutions to Theoretical Questions:

## Cloud Computing Basics

1. In what ways does deploying an application in the cloud differ from deploying it on-premises, particularly regarding security, deployment speed, and resource management?

Use examples to illustrate your points.

Deploying an app in the cloud versus running it on-premises comes with some big differences—especially when it comes to security, how fast you can launch, and managing resources. Let’s break it down with some simple examples, like your EventEase Booking App hosted on Azure.

**Security**

* In the cloud, like on Azure, security is a shared job. Microsoft handles things like the physical safety of their servers and keeping the infrastructure secure. Meanwhile, you focus on making sure your app is protected—like managing who can log in or securing user data.
  + *Example: You can use Azure Active Directory to manage logins, set up database firewalls, and turn on HTTPS to encrypt traffic—without needing to worry about physical server access.*
* On-premises, it’s all on you. You’re in charge of everything—from locking the server room to installing antivirus software and keeping the operating system updated.
  + *Example: If a university hosted the EventEase app on-site, their IT team would need to secure the physical servers, manage backups, and set up all security protocols themselves.*

**Deployment Speed**

* Cloud deployment is much faster. You can push updates or rollbacks with just a few clicks or through automated pipelines using tools like GitHub Actions or Azure DevOps.
  + *Example: For EventEase, you could update the site directly from your laptop and see the changes live in minutes.*
* On-premises setups take way longer. You might need to order and install hardware, configure the servers, and test everything manually.
  + *Example: Getting the EventEase app live from scratch in a local server room could take weeks.*

**Resource Management**

* In the cloud, you only use (and pay for) what you need. If your app suddenly gets more traffic—like during a popular event—Azure can automatically give you more power to keep things running smoothly.
  + *Example: If lots of users are booking venues at once, Azure can scale up your resources without you doing a thing.*
* On-premises, you’re limited by whatever equipment you’ve got. If you need more power, you have to buy and set up new hardware, which can be slow and expensive.
  + *Example: To prepare for peak season, an on-prem setup might need to over-buy equipment just in case—most of which won’t be used the rest of the year.*

2. What are the key differences between Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), and why might EventEase benefit from the use of PaaS over the other two when building a new application? Use examples to support your answer.

**What’s the Difference?**

|  |  |  |  |
| --- | --- | --- | --- |
| Service Model | What You Manage | What the Provider Manages | Example |
| IaaS | You manage everything *except* the physical infrastructure—OS, apps, runtime, etc. | The cloud provider gives you the virtual machines, storage, and networking | Azure Virtual Machines, AWS EC2 |
| PaaS | You focus on just the app code and data | The provider handles everything else—servers, runtime, database, OS, scaling, etc. | Azure App Service, Google App Engine |
| SaaS | You just use the software | Everything (infrastructure + app) is managed by the provider | Microsoft 365, Gmail, Zoom |

**So, Why Should EventEase Use PaaS?**

When building a new app, like the EventEase Booking System, using Platform as a Service (PaaS) is often the sweet spot. Here’s why:

**1. Less Maintenance, More Focus on Features**

* With PaaS, your team doesn’t have to worry about managing servers, applying OS updates, or configuring load balancers.
* You can spend more time adding cool features—like user dashboards, payment integration, or event reminders.

*Example:* With Azure App Service, you can deploy your .NET Core EventEase app, connect it to an Azure SQL Database, and let Azure handle performance, patching, and availability in the background.

**2. Built-in Tools for Developers**

* PaaS platforms come with helpful tools like auto-scaling, deployment slots, diagnostics, and CI/CD integration out of the box.
* These features help speed up development and make your app more reliable.

*Example:* If EventEase is expecting more traffic during graduation season, Azure can automatically scale up resources to handle more bookings—no manual work needed.

**3. Better for Fast Development & Scaling**

* PaaS supports rapid prototyping and agile development. You can launch updates often, test easily, and grow as needed without rethinking your infrastructure.

*Example:* If you want to add an analytics dashboard for venue owners on EventEase, you can quickly build and test it without worrying about hardware capacity.

# ERD Diagram

# 

# Screenshots of the web application

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# Reference List

Microsoft Corporation, 2022. *ASP.NET Core MVC Documentation*. Available at: <https://docs.microsoft.com/en-us/aspnet/core/mvc/> [Accessed 18 March 2025].

Microsoft Corporation, 2022. *Entity Framework Core Documentation*. Available at: <https://docs.microsoft.com/en-us/ef/core/> [Accessed 26 March 2025].

Microsoft Azure, 2022. *Azure SQL Database*. Available at: <https://azure.microsoft.com/en-us/services/sql-database/> [Accessed 4 April 2025].

Stack Overflow, 2022. *Common ASP.NET Core MVC Errors and Solutions*. Available at: <https://stackoverflow.com/questions/tagged/asp.net-core-mvc> [Accessed 7 April 2025].