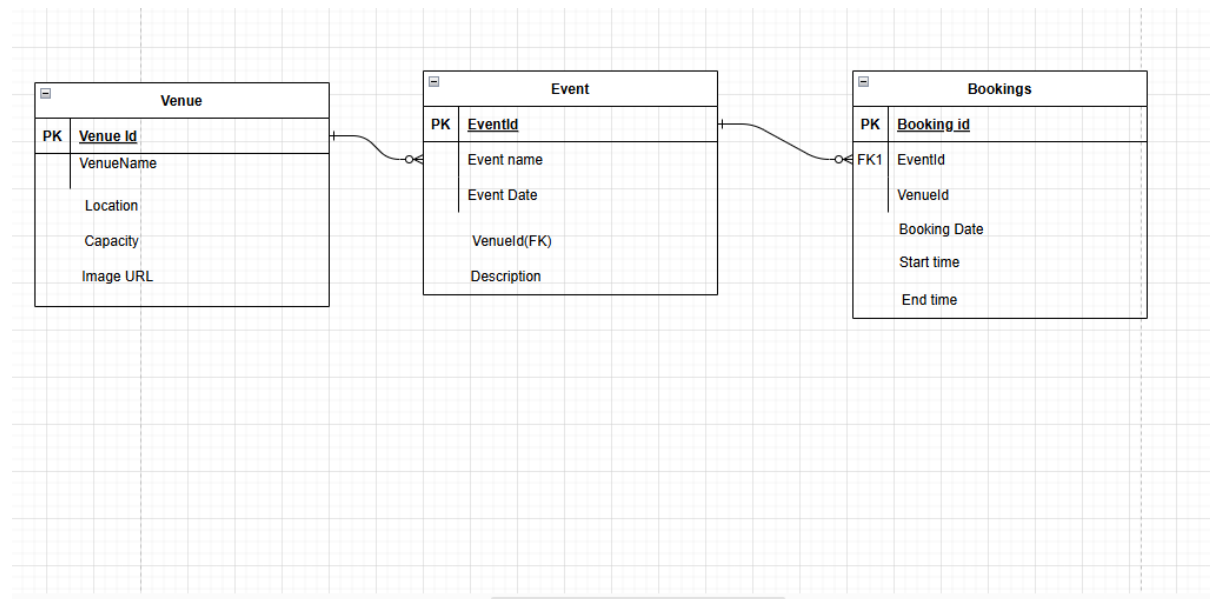


## CLOUD DEVELOPMENT POE PART 1



### Documentation

The screenshots bellow are documentation for the following requirements. First being C: Deploy your webapp and database to Microsoft azure. Screenshot of the service created in the portal , successfully deployment

Home > Microsoft.SQLDatabase.newDatabaseNewServer\_6d2d67371ea34c8dbdb90 | Overview

Deployment

Search [ ] x << [ ] Delete [ ] Cancel [ ] Redeploy [ ] Download [ ] Refresh [ ]

Overview

Inputs

Outputs

Template

**✓ Your deployment is complete**

Deployment name : Microsoft.SQLDatabase.newDatabase... Start time : [ ]

Subscription : ADVTECH-Tertiary Vega School Correlation ID : 50afe39a-b60f-410e-ac4e-b239ea93e...

Resource group : AZ-JHB-RSG-IIEMSA-ST10251297-TER

> Deployment details

✓ Next steps

[Go to resource](#)

Give feedback

[Tell us about your experience with deployment](#)

**Deployment succeeded**

Deployment 'Microsoft.SQLDatabase.newDatabaseNewServer\_6d2d...' to resource group 'AZ-JHB-RSG-IIEMSA-ST10251297-TER' was successful.

[Go to resource](#) [Pin to dashboard](#)

**Cost management**

Get notified to stay within your budget and prevent unexpected charges on your bill.

[Set up cost alerts >](#)

**Microsoft Defender for Cloud**

Secure your apps and infrastructure

[Go to Microsoft Defender for Cloud >](#)

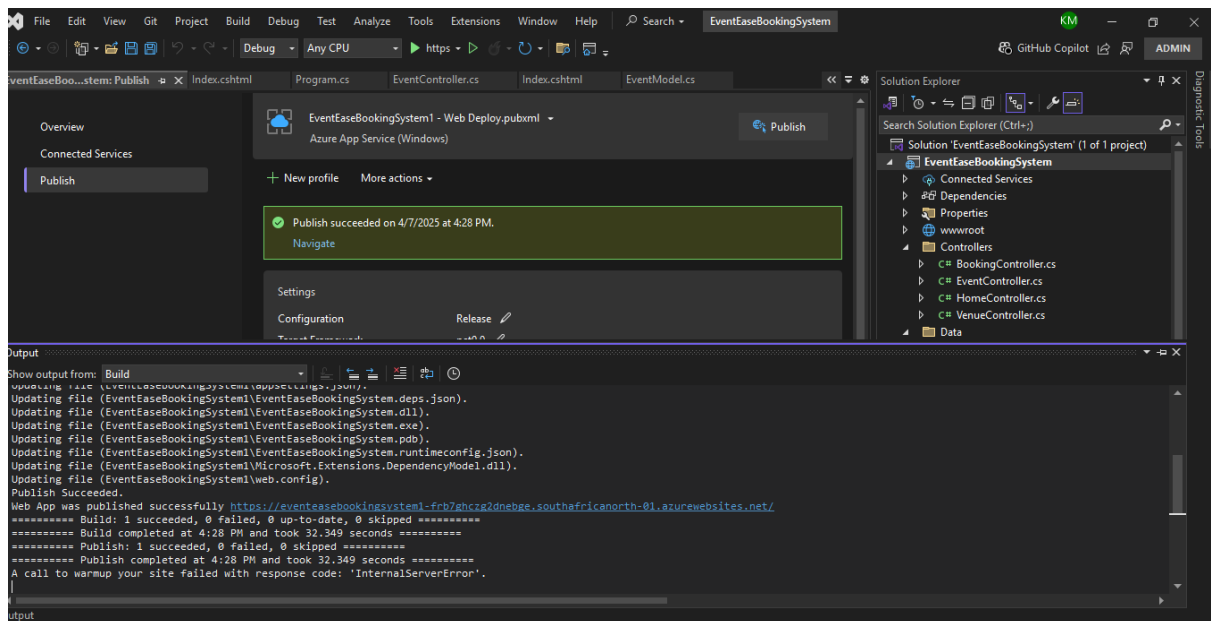
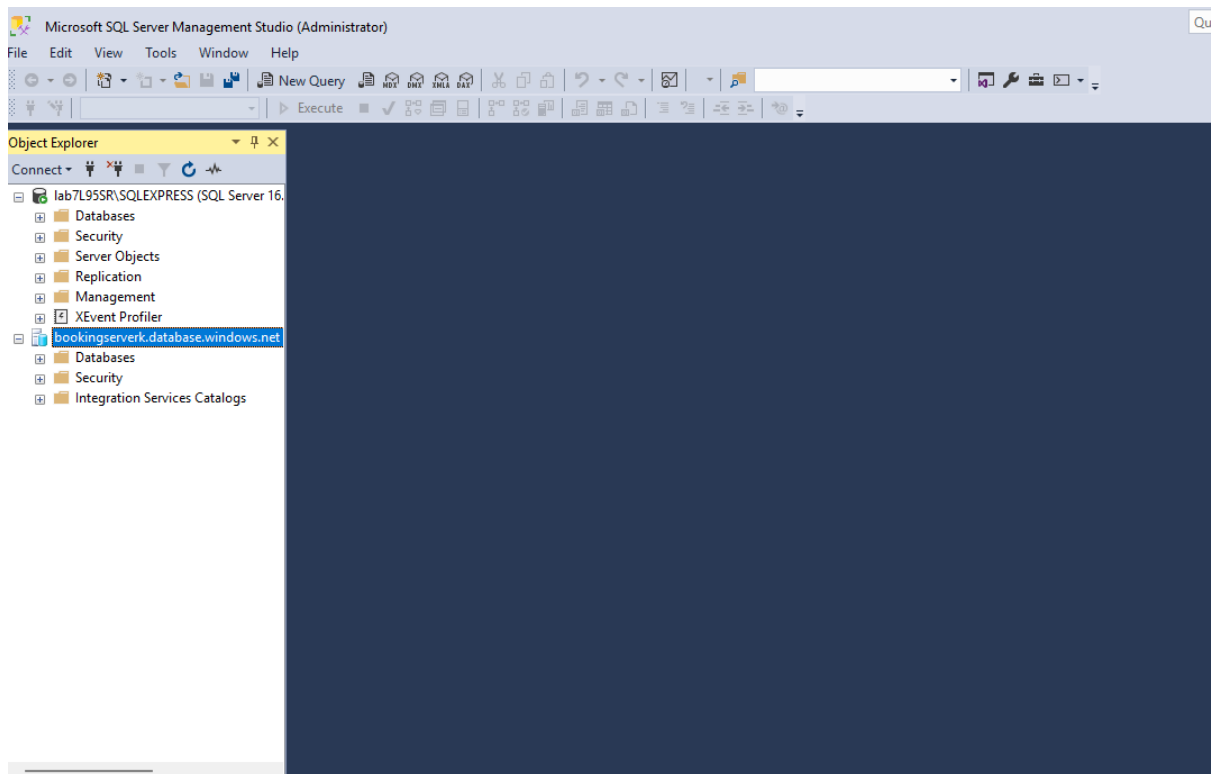
**Free Microsoft tutorials**

[Start learning today >](#)

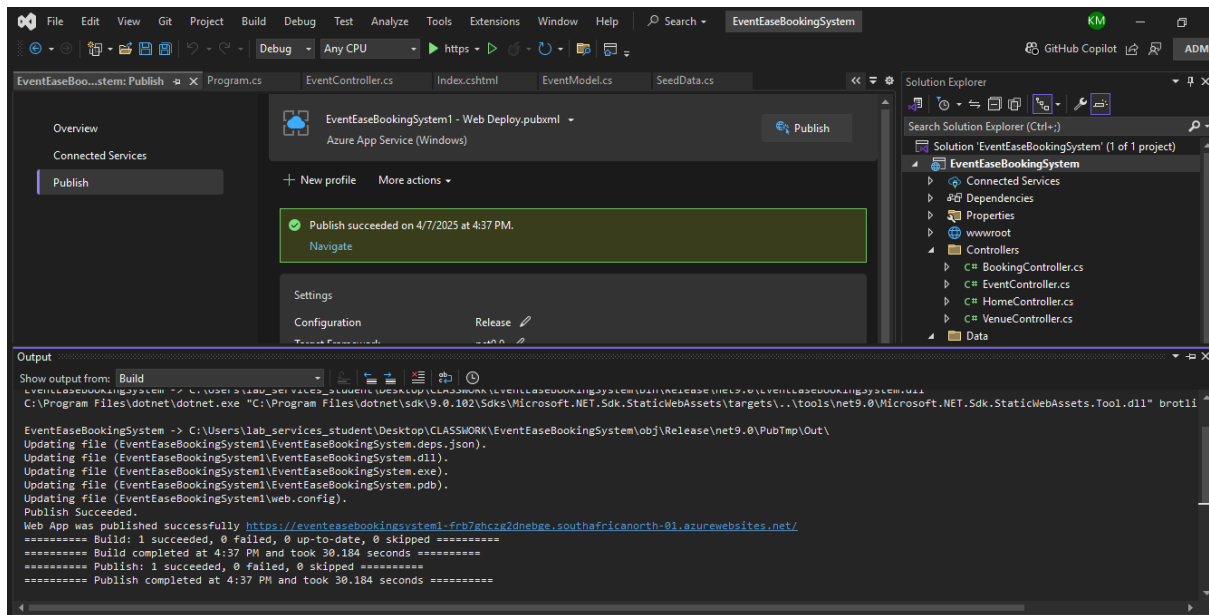
**Work with an expert**

Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.

2. Screenshot of the SQL database created in the portal or migrated



3. Screenshot of the program in Visual Studio code being published to Microsoft Azure



BELOW ARE THE LINKS

<https://eventeasebookingsystem1-frb7ghczg2dnebg.e.southafricanorth-01.azurewebsites.net/>

**Event ease booking system:**

<https://eventeasebookingsystem1-frb7ghczg2dnebg.e.southafricanorth-01.azurewebsites.net/>

## Theoretical

1. **On-premises installation and installation within the cloud are two different approaches to handling and running software or services with their own strengths and trade-offs. Following is an overview of the differences:**

### 1. Infrastructure

**On-Premises:** All hardware, servers, and infrastructure are present and hosted within the physical organizational framework. The organization is responsible for purchasing, sustaining, and upgrading this infrastructure.

**Cloud:** Third-party provider hosts and maintains the hardware (e.g., Microsoft Azure, AWS, Google Cloud). The resources are accessed via the web, with maintenance and scalability handled by the provider.

### 2. Costs

**On-Premises:** Typically higher upfront costs for hardware, software, and setup, with additional ongoing costs for maintenance and upgrades.

**Cloud:** Usually pay-as-you-go, lower up-front costs. Only pay for resources and

services consumed, which is often less expensive in the long term.

### 3. *Scalability*

On-Premises: Limited by the physical infrastructure. Scale up by needing to buy extra hardware and perhaps downtime to set up.

Cloud: Highly scalable. You can rapidly scale up or down on-demand, usually in a couple of clicks, without considering physical constraints.

### 4. *Control*

On-Premises: Gives complete control of infrastructure, security, configuration, and data processing.

Cloud: Control is shared with the cloud provider. While you can control most of the features of the services, you must trust the provider with fundamental security and compliance.

### 5. *Security*

On-Premises: All security is handled internally, which might be advantageous for organizations with high compliance needs.

Cloud: Cloud providers invest heavily in security measures, but your data resides on their servers, which might be a concern for some sensitive industries.

### 6. *Flexibility*

On-Premises: Less flexible in terms of mobility. Access is generally limited to on-site or VPN setups.

Cloud: Offers high flexibility and accessibility from anywhere with an internet connection, making it ideal for remote or distributed teams.

### 7. *Maintenance*

On-Premises: All updates, patches, and troubleshooting are handled by the IT staff.

Cloud: Updates and maintenance are handled by the provider, reducing the workload for in-house IT personnel.

The choice between the two depends on the organization's specific requirements, budget, and priorities. For example, organizations with very sensitive data would opt for on-premises, while those requiring agility and cost savings might opt for the cloud.

## 2. **Identify key differences between Azure hosting models.**

Azure offers a variety of hosting models, each appropriate for varying application needs. Listed below are the key distinctions among some of the most used Azure hosting

models:

#### 1. Azure Virtual Machines (VMs)

What it is: Provides total control of a virtualised server environment.

Use case: Good for applications that require customized setups or older applications that have to be relocated with no modifications.

Management: You are on your own with operating system, updates, and software.

#### 2. Azure App Service

What it is: A fully managed platform for hosting web apps, APIs, and mobile backends.

Use case: Best suited for developers who have to focus on application development without worrying about infrastructure.

Management: Scaling, patching, and maintenance are taken care of by Azure.

#### 3. Azure Kubernetes Service (AKS)

What it is: A managed Kubernetes service used to deploy and manage containerized applications.

Use case: Best suited for microservices architecture and containerized workloads.

Management: Azure handles the Kubernetes control plane, and you handle applications.

#### 4. Azure Functions

What it is: A serverless computing platform for running event-driven code.

Use case: Most suitable for lightweight, short-lived event-driven tasks.

Management: Fully managed by Azure, dynamically scaled.

#### 5. Azure Container Instances (ACI)

What it is: A container service that runs without virtual machine management.

Use case: Best for small, standalone containerized workloads.

Management: Underlying infrastructure managed by Azure.

#### 6. Azure Service Fabric

What it is: Distributed systems platform for developing scalable and fault-tolerant applications.

Use case: Best for complicated, stateful apps that need high availability.

Management: You own the application, and Azure offers the platform.

Each hosting model has its advantage, and the decision relies on considerations such as application design, scalability requirements, and management style