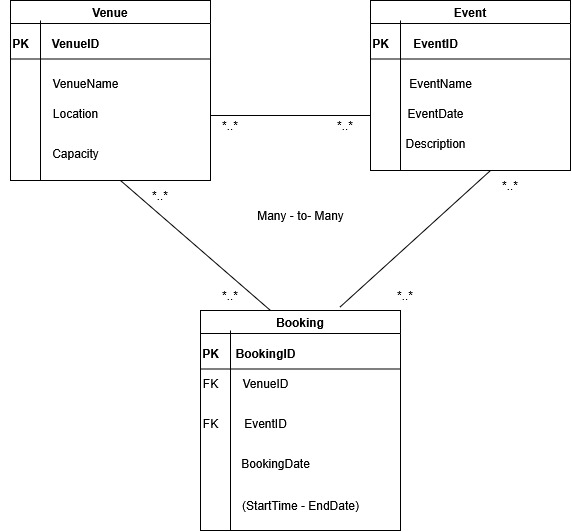
Part 1

1. **Design the EventEase Database**

****

CREATE DATABASE EventEaseDB;

USE EventEaseDB;

CREATE TABLE Venue (

VenueId INT IDENTITY(1,1) PRIMARY KEY,

VenueName VARCHAR(255) NOT NULL,

VenueLocation VARCHAR(255) NOT NULL,

Capacity INT NOT NULL,

ImageURL VARCHAR(255) UNIQUE

);

CREATE TABLE Event (

EventId INT IDENTITY(1,1) PRIMARY KEY,

EventName VARCHAR(255) NOT NULL,

EventDate DATE NOT NULL,

EventDescription VARCHAR(MAX) NOT NULL

);

CREATE TABLE Booking (

BookingId INT IDENTITY(1,1) PRIMARY KEY,

EventId INT NOT NULL,

VenueId INT NULL,

BookingDate DATE NOT NULL,

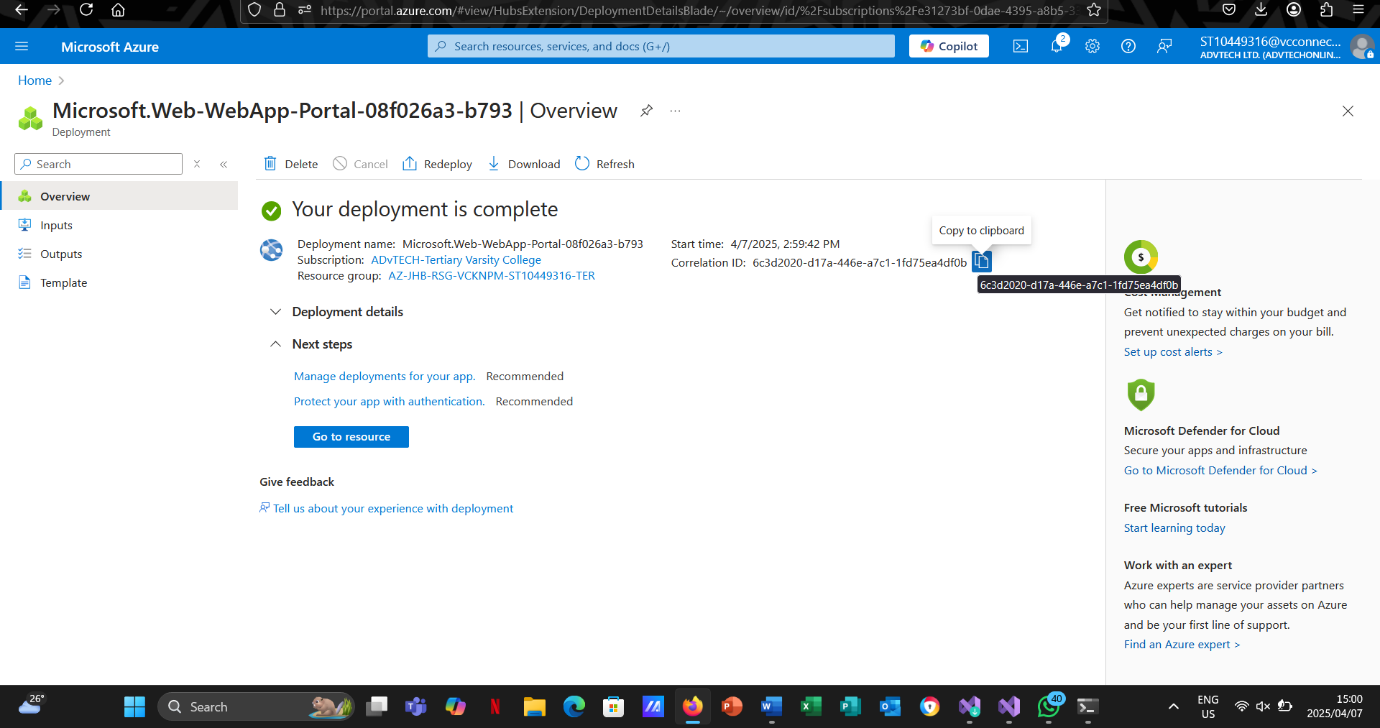
FOREIGN KEY (EventId) REFERENCES Event(EventId) ON DELETE CASCADE,

FOREIGN KEY (VenueId) REFERENCES Event(VenueId) ON DELETE CASCADE,

UNIQUE (VenueId , VenueId )

);

1. **Developing the Web Application**

****

**A screenshot of a computer

Description automatically generated**

**C. Deploying Web Application and Database to Microsoft azure**

**A screenshot of a computer

Description automatically generated**

A screenshot of a computer

Description automatically generated

Website Link: <https://st10449316eventease-abfbhhgufpewcqcy.eastus-01.azurewebsites.net/>

GitHub Link: <https://github.com/Modies20/EventsEase.git>

**D. Cloud Computing Basics**

**1. Differences Between Cloud Deployment and On-Premises Deployment**

Instead of installing and maintaining those resources on-premises, when you deploy an application in the cloud, you are essentially renting computing resources (servers, storage, networking, etc.) from a third-party provider like Google Cloud Platform (GCP), Microsoft Azure, or Amazon Web Services (AWS). Several significant changes become apparent, especially with regard to resource management, development speed, and security.

A) Security

* Shared Responsibility Model: Cloud providers generally follow a shared responsibility model in which the customer (you) oversees protecting the apps, data, and configurations that run on the underlying infrastructure, while the cloud provider is in charge of protecting the infrastructure itself (AWS, n.d.-a). On-premises, your internal teams are fully responsible for all security duties (physical, network, and application).
* Physical Security: Cloud companies maintain several geographically separated data centers which have disaster recovery plans, strict physical access controls, and fire suppression. Local security measures are used in on-premises situations, and their impact differs widely based on resources and level of experience (NIST, 2011).
* Scalable Security Services: A lot of cloud platforms come with integrated security features like threat detection, encryption key management, identity and access management (IAM), and more. Similar restrictions can be established via on-premises solutions, but they usually need specialized staff, manual setup, and greater upfront expenses.

B) Development speed

* Rapid Provisioning: Without having to wait for the purchase and installation of new hardware, developers can immediately spin up virtual machines, containers, or serverless functions in the cloud (Microsoft Azure, n.d.). Lead times are shortened from weeks or months to just a few minutes or hours as a result.
* Continuous Integration/Continuous Deployment (CI/CD) and DevOps: Cloud platforms usually provide managed CI/CD pipelines, which support quicker deployment and iteration. Similar pipeline setup and maintenance can be more difficult and time-consuming on-premises.
* Emphasis on Innovation: Development teams may concentrate more on writing code and developing new features since cloud providers take care of a large portion of the basic heavy lifting (such as server maintenance and networking setup).

C) Resource Management

* Elasticity and Scalability: Cloud services let you adjust the size of resources as needed (AWS, n.d.-b). You only pay for what you use thanks to its elasticity. However, getting gear for on-premises deployments costs a large capital investment (CapEx) that might go unused.
* Cost Structure: Pay-as-you-go (OpEx) is the typical cloud cost model, which allows experimentation and scaling without requiring significant upfront expenses. The initial and continuing maintenance costs of on-premises systems are higher.
* Automation and Orchestration: Cloud environments usually incorporate infrastructure-as-code services (like AWS CloudFormation and Azure Resource Manager) and orchestration tools (like Kubernetes for containers). Although these technologies can be used with on-premises solutions, setup is frequently more time-consuming and needs specific knowledge

For instance, a startup wishing to create a web application can easily set up a serverless function or virtual server on AWS, deploy the code, and test it in a matter of hours. Before even starting application testing in an on-premises setting, the same startup may have to wait weeks to acquire a physical server, set up an operating system, and set up network and security rules.

**2. Differences Between IaaS, PaaS, and SaaS and Why EventEase Might Benefit from PaaS**

A) IaaS, or infrastructure as a service

Definition: Infrastructure as a Service (IaaS) offers the basic components—networks, storage, and virtual machines—through the internet. Installing and maintaining your own operating systems, software and apps on top of this infrastructure is your responsibility (IBM Cloud, n.d.).

For instance, Google Compute Engine (Google Cloud), Microsoft Azure Virtual Machines, and Amazon EC2 (AWS).

Use Cases: Ideal for groups with difficult, customized settings or those requiring smooth control over their environment.

B) PaaS, or platform as a service

Definition: PaaS offers an environment that lets you create, perform, and maintain applications without having to worry about the underlying infrastructure. This environment includes an operating system, realtime environment, development frameworks, and more (Red Hat, n.d.).

Examples would be Microsoft Azure App Service, Google App Engine, and AWS Elastic Beanstalk.

Use Cases: Because it abstracts away the specifics of server management and frequently connects with CI/CD pipelines, it is perfect for rapid development and deployment. Instead of setting up servers, developers can concentrate on writing code.

C) SaaS, or software as a service

Definition: SaaS (Salesforce, Office 365, Google Workspace, etc.) offers ready for use software programs that are sold online on a subscription or usage basis. The end user does not control the application code or the infrastructure; they just configure the software.

Use cases include CRM systems and productivity suites, when you require a fully functional application free of management overhead.

**The Potential Benefits of PaaS for EventEase**

* Faster Time-to-Market: Without having to set up or maintain underlying infrastructure, PaaS provides EventEase to focus on developing and improving its core event management features. New features may be released more quickly as a result.
* Integrated Scalability: PaaS solutions frequently offer automatic scaling in response to application demand, which is essential for an events platform that could see significant traffic spikes during important events.
* Reduce Operational Overhead: PaaS removes operations teams of the load of managing servers by abstracting it away, allowing them to concentrate on user experience, security, and certification rather than on regular server maintenance.
* Integrated Services: A lot of PaaS solutions come with messaging wait times, databases, analytics, and other interfaces that EventEase can use to create strong features faster (AWS, n.d.-c; Microsoft Azure, n.d.).

On the other hand, SaaS might not be adaptable enough for a custom event management application if an off-the-shelf solution doesn't satisfy EventEase's particular needs, and IaaS would require more manual setup and maintenance of the operating system and runtime environments.

**Referencing**

**AWS (n.d.-a), *Shared Responsibility Model*. Available at:** [**https://aws.amazon.com/compliance/shared-responsibility-model/**](https://aws.amazon.com/compliance/shared-responsibility-model/) **(Accessed: 1 April 2025).**

**AWS (n.d.-b), *Amazon EC2*. Available at:** [**https://aws.amazon.com/ec2/**](https://aws.amazon.com/ec2/) **(Accessed: 2 April 2025).**

**AWS (n.d.-c), *AWS Elastic Beanstalk*. Available at:** [**https://aws.amazon.com/elasticbeanstalk/**](https://aws.amazon.com/elasticbeanstalk/)

**(Accessed: 2 April 2025).**

**IBM Cloud (n.d.), *IaaS, PaaS, and SaaS Explained*. Available at:** [**https://www.ibm.com/cloud/learn/iaas-paas-saas**](https://www.ibm.com/cloud/learn/iaas-paas-saas)

**(Accessed: 4 April 2025).**

**Microsoft Azure (n.d.), *Azure App Service*. Available at:** [**https://azure.microsoft.com/en-us/products/app-service/**](https://azure.microsoft.com/en-us/products/app-service/)

**(Accessed: 5 April 2025).**

**NIST (2011), *The NIST Definition of Cloud Computing (SP 800-145)*. Available at:** [**https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf**](https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf)

**(Accessed: 5 April 2025).**

**Red Hat (n.d.), *What is PaaS (Platform as a Service)?* Available at:** [**https://www.redhat.com/en/topics/cloud-computing/what-is-paas**](https://www.redhat.com/en/topics/cloud-computing/what-is-paas) **(Accessed: 6 April 2025).**