

Car Rental Solution Deployment Plan

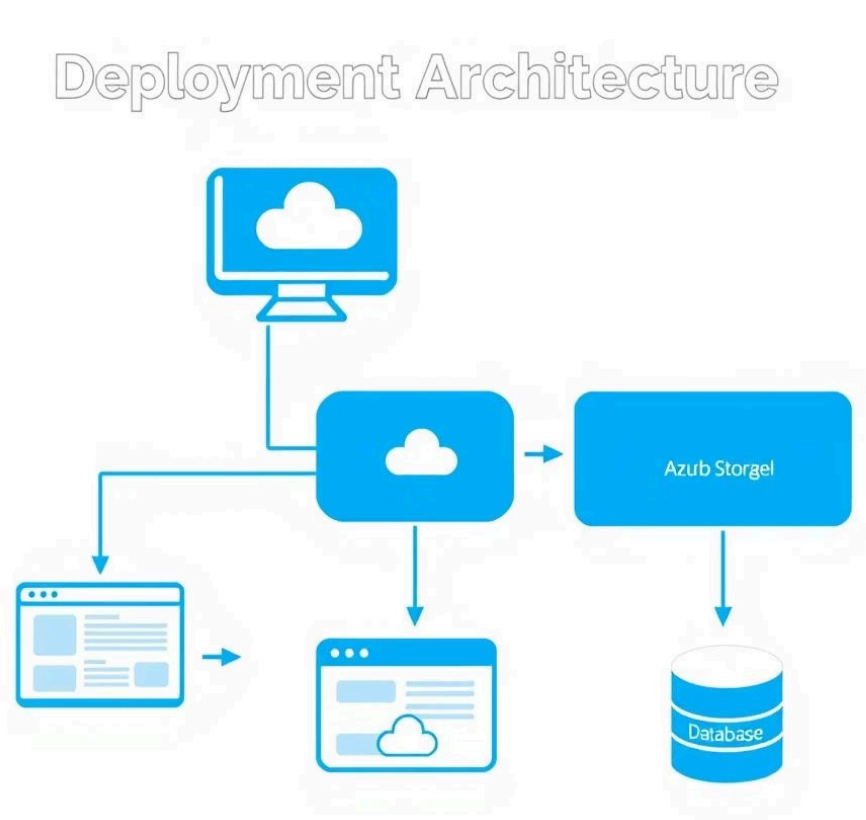
This document outlines the deployment plan for the car rental solution, detailing the architecture, deployment options, and step-by-step instructions for a successful launch.

Solution Architecture

- Admin Panel:** ASP.NET MVC application with an SQL Server database. This includes the main application URL and a separate API URL, requiring two distinct deployments.
- Azure Blob Storage:** Utilized for storing documents and images, ensuring scalable and cost-effective storage.
- Frontend Website:** A React.js application that communicates with the API to handle booking data and user interactions.

Deployment Options

The application can be deployed on Azure App Service or using alternative hosting solutions. Here's a more detailed deployment plan.



Detailed Deployment Plan

- Prerequisites:**
 - An active Azure subscription or alternative hosting environment.
 - SQL Server instance (Azure SQL Database recommended).
 - Azure Blob Storage account.
 - Visual Studio for ASP.NET MVC deployment.
 - Node.js and npm for React.js deployment.
- Admin Panel Deployment:**
 - Deploy the ASP.NET MVC application to Azure App Service (or preferred hosting).
 - Configure connection strings to point to the SQL Server database.
 - Deploy the API URL separately, ensuring it is secured and accessible.
 - Test both application and API endpoints thoroughly.
- Azure Blob Storage Configuration:**
 - Create containers within the Azure Blob Storage account for documents and images.
 - Configure appropriate access policies and security settings.
 - Update the Admin Panel to use Azure Blob Storage for uploading and retrieving files.
- Frontend Website Deployment:**
 - Build the React.js application for production.
 - Deploy the built files to Azure App Service (or preferred hosting).
 - Configure the frontend to communicate with the deployed API URL.
 - Ensure all frontend routes and functionalities are working correctly.
- Testing and Validation:**
 - Perform end-to-end testing to verify the entire solution is working as expected.
 - Conduct user acceptance testing (UAT) with key stakeholders.
 - Monitor application performance and address any issues promptly.
- Go-Live and Monitoring:**
 - Coordinate the go-live date and communicate with all relevant parties.
 - Continuously monitor application health, performance, and security.
 - Establish a process for addressing bugs and feature requests.