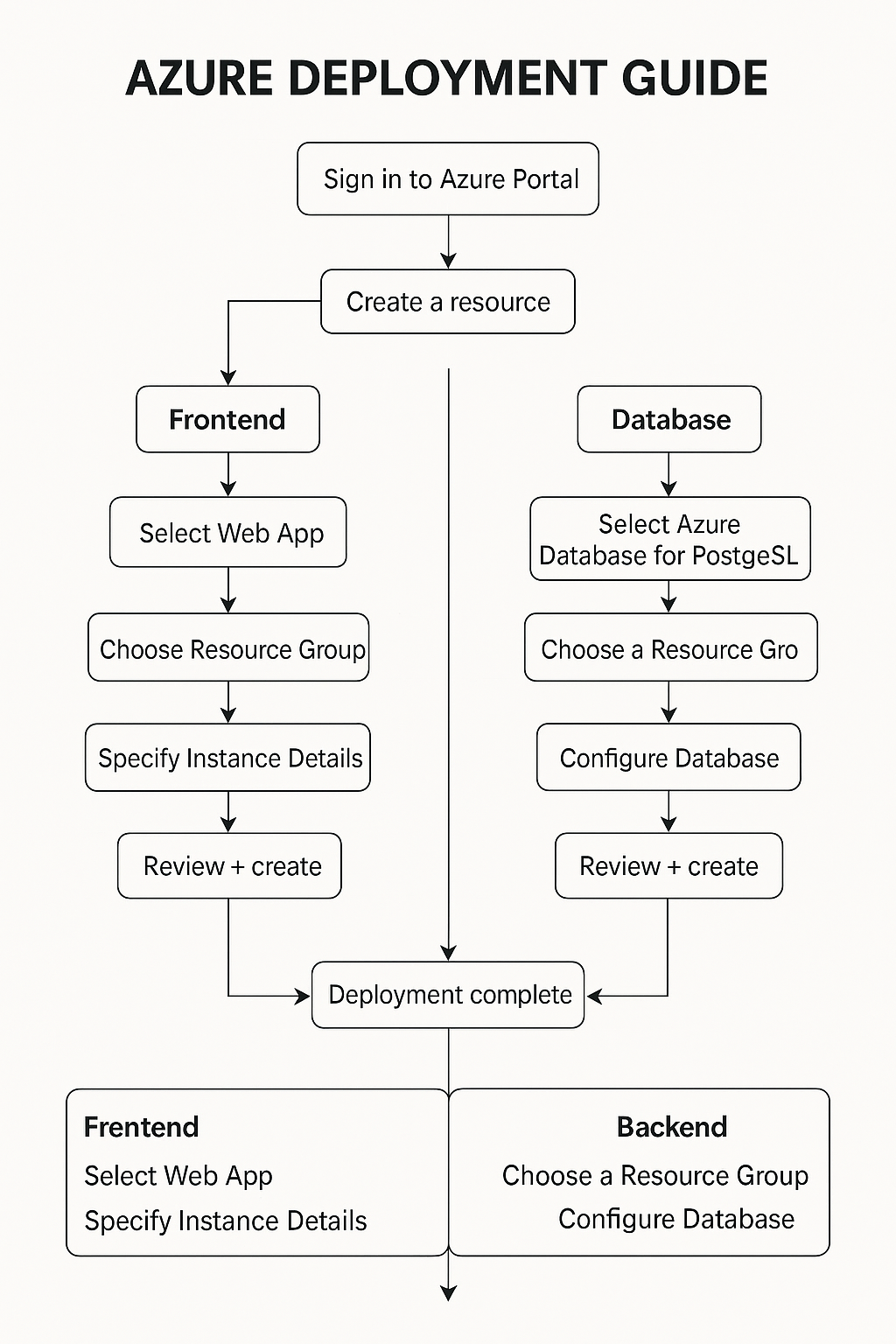
End-to-End Azure Deployment Guide for Ecommerce Prototype



# 1. Prerequisites

* - An active Microsoft Azure account (https://azure.microsoft.com)
* - Source code for: Frontend, Backend, SQL scripts/schema
* - Azure CLI installed (optional)

# 2. Create a Resource Group

* 1. Login to Azure portal: https://portal.azure.com
* 2. Click 'Resource groups' > '+ Create'
* 3. Fill in Subscription, Resource Group Name, Region
* 4. Click 'Review + Create' > 'Create'

# 3. Deploy the Database

* 1. Search for 'Azure Database for MYSQL > '+ Create'
* 2. Select Flexible Server
* 3. Configure Server name, Region, Authentication, Admin credentials
* 4. Networking: Public access, allow IPs
* 5. Click 'Review + Create' > 'Create'
* 6. Use Query Editor or pgAdmin to create tables

# 4. Deploy the Backend (e.g., FastAPI) using App Service

* 1. Search 'App Services' > '+ Create'
* 2. Fill in Name, Runtime stack, Region
* 3. App Service Plan: Create New, use Free tier
* 4. Deploy via GitHub, Local Git, or Zip deploy
* 5. Set environment variables in Configuration

# 5. Deploy the Frontend (e.g., React App) using Static Web Apps

* 1. Search 'Static Web Apps' > '+ Create'
* 2. Configure Name, Region, Deployment source (GitHub/manual)
* 3. Build details: App location, Output location
* 4. Deploy and access via provided URL

# 6. Connect Frontend, Backend, and Database

* - Update frontend with backend API URL
* - Backend uses environment variables for DB access
* - Test integration

# 7. Monitoring and Logging

* Navigate to your **App Service** in the Azure Portal.
* In the left-hand menu under **Settings**, click on **Application Insights**.
* Click Turn on Application Insights.
* If you don’t already have an Application Insights resource, choose **Create New**, then select a region and a name.
* Select or create a **Log Analytics Workspace** when prompted.
* Click **Apply** to enable monitoring.
* After setup, navigate to the **Application Insights** resource:
* Use **Live Metrics Stream** for real-time data.
* Explore **Failures**, **Performance**, and **Availability** for diagnostics.
* For more advanced analysis, go to your **Log Analytics Workspace** and run Kusto queries to explore collected logs.

# 8. Secure Your Application

* Go to Azure Portal and search for **Key Vault**.
* Click **+ Create** and fill in the required details (name, region, pricing tier).
* Once created, navigate to your Key Vault > **Secrets** > **+ Generate/Import**.
* Add a new secret for credentials like your DB password or API keys.
* Go to your **App Service** > **Configuration** > **Application settings**.
* Click **+ New application setting**, then use the syntax:  
  @Microsoft.KeyVault(SecretUri=https://<YourKeyVaultName>.vault.azure.net/secrets/<SecretName>/)  
  Replace the placeholders accordingly.
* To enforce HTTPS:
* Go to **TLS/SSL settings** under your App Service.
* Set **HTTPS Only** to **On**.
* To configure CORS settings:
  + In the App Service, go to **API** > **CORS**.
  + Add allowed origins (e.g., https://your-frontend.azurestaticapps.net).

# Appendix

* - Azure Portal: https://portal.azure.com
* - Azure CLI: https://learn.microsoft.com/en-us/cli/azure/install-azure-cli
* - App Service: https://learn.microsoft.com/en-us/azure/app-service/
* - Static Web Apps: https://learn.microsoft.com/en-us/azure/static-web-apps/
* - PostgreSQL: https://learn.microsoft.com/en-us/azure/postgresql/