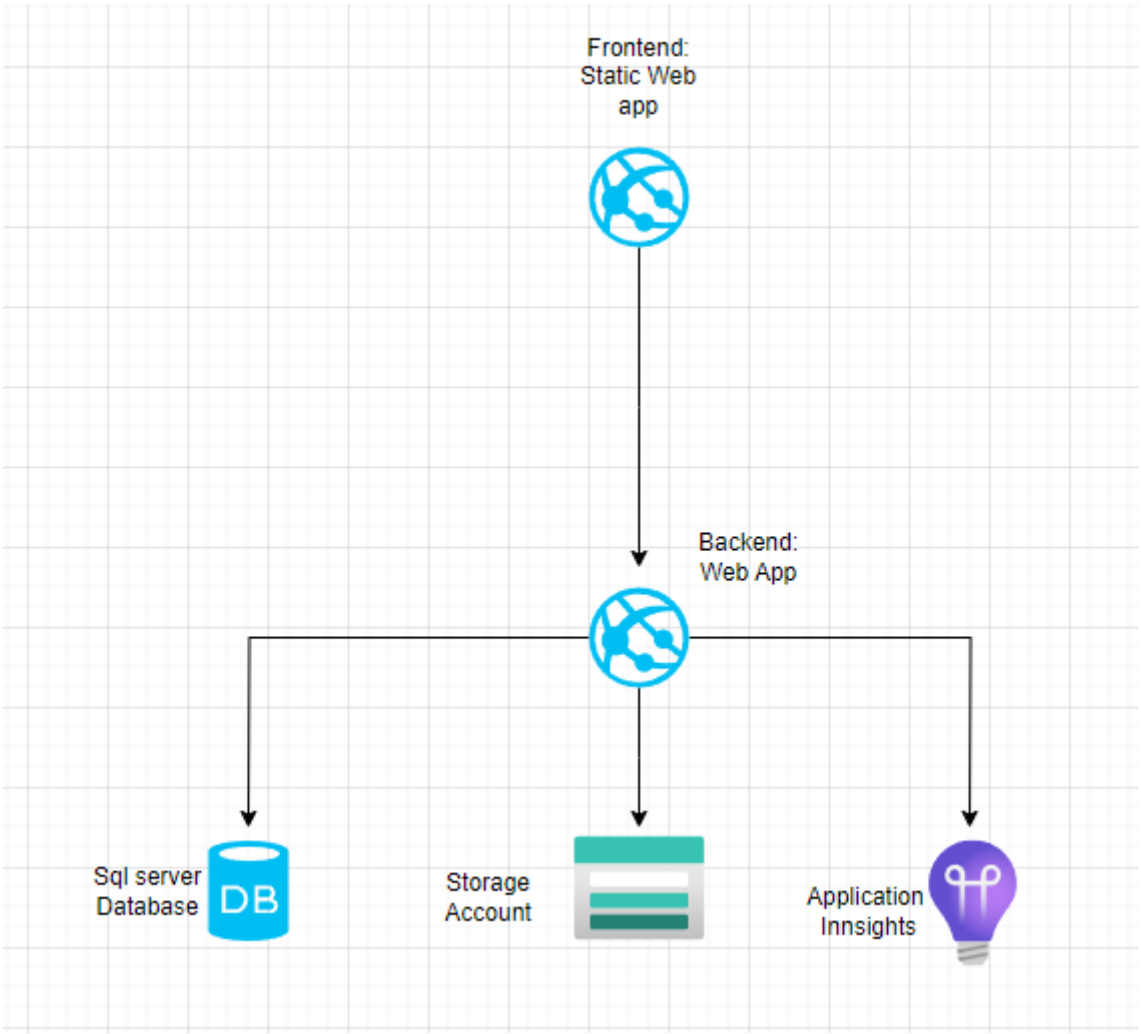
Draw a diagram and explain what Azure services are best to use in serving the APIs to a FE.
For a smaller project, I would host an API serving a Frontend I would use Azure App service for the API and a static web app for the frontend. Which is ideal for hosting Web apps and APIs. App service is a great option when you need a fully managed platform to deploy web applications. For a relatively simple application I would go with something like this:



But for a larger project I would use add Azure Container Registry to store images for the Frontend and Backend And pull the lastest images to the web application instead. In this case the Frontend would also be hosted in a App service aswell.

What services should you use for handling access management and log-in?
There are many different options, you could go for Microsoft Entra ID, which has seamless integration with .NET and can be used to implement authentication and securing APIs.
You can also use OpenId connect for authentication and OAuth 2.0 framework for Athorization.

What technologies can you use to deploy this service to cloud?
There are many options to deploy the services to the cloud. To deploy this solution to the cloud I would use Azure DevOps pipelines.
I would create two separate repositories: One for deploying the infrastructure and one for deploying the code to the services.
For deploying the infrastructure you can use Terraform/Bicep to write Infrastructure as code (IaC) to

automate the deployment of infrastructure. This way the development process will be more effective and consistent while heavily reducing manual errors and operational costs.

The second pipeline would be used to deploy changes to the code. For the example I gave above with images, you would need a simple pipeline for the deployment:
In the first stage:
Build and push images to Azure Container Registry.
And in the next stage if the first one was successful, pull the latest images from ACR to their respective web apps.

With previous images of the applications in ACR, version control and rollbacks becomes very easy to perform. For the pipelines to perform as intended, your services needs access to certain areas. Using service connections and role-based access control ensures that access is tightly controlled. For example the web apps would need AcrPull rights from ACR, to pull the images.