Application Insight monitoring.

Goal-

1. Deploy a api application page which stream a video. Video stored on storage account blob container. And the application hosted on server 3.
2. Deploy another application which display a table of product by fetching data from azure sql.

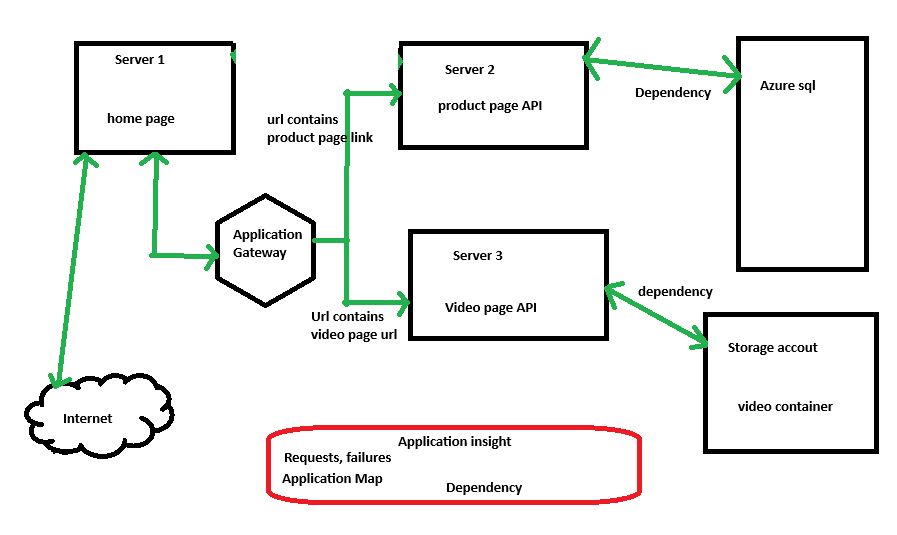
This application is hosted on server 2.

1. Integrate application insight with both applications. Product and video. Test application requests from azure application insight.
2. Do corruption on video container or in database to test application dependency failures.
3. Create a home page with contains two options one is to see product another one to see video. Deploy on server 1.

While click on video button it should redirect to video server

While click on product it goes to product server.

1. Implement application gateway to route traffic based on url.
2. Integrate application insight to home page on server 1 to monitor entire workflow.



Task 1.

Create storage account and upload a video to that.

A screenshot of a computer

Description automatically generated

Check video accessibility

A screenshot of a cartoon cat and a mouse

Description automatically generated

Task 2.

Create an application to watch video.

<body>

    <main>

        <h2>Tom and Jerry</h2>

        <video controls>

            <source src="https://videostracc.blob.core.windows.net/my-video-container/Tom-Jerry.mp4" type="video/mp4">

            Your browser does not support the video tag.

        </video>

    </main>

</body>

Check video application working functionality by running locally.

A cartoon of a cat

Description automatically generated

Task 3.

Create application insight to integrate with application to monitor.

Note- while creating Application insight we need to either to create a new Workspace or to attach if we already have one.

Reason –

**Workspace as a Data Store**: The Log Analytics workspace is used to store telemetry data that is collected from your Application Insights resource. This includes performance data, exceptions, requests, dependencies, custom events, and more.

** Application Insights as Data Collector:**

* **Yes, Application Insights is used to collect telemetry data from your application. This includes things like performance metrics, request and response times, exceptions, dependencies, custom events, etc. It acts like an agent (through SDKs or through integration with services) that gathers data from the application running in different environments (e.g., web apps, mobile apps, etc.).**

** Workspace and Data Storage:**

* **Yes, the telemetry data collected by Application Insights is stored in a Log Analytics workspace. The workspace serves as the central storage for this telemetry, and it allows you to query and analyze the data using Kusto Query Language (KQL).**

**A screenshot of a computer

Description automatically generated**

**Task 4.**

Integrate application insight to application to test this I am creating an angular application as welcome page.

Will host it on app service and link to video server.  
  
create video server

A screenshot of a computer

Description automatically generated

**A screen shot of a computer program

Description automatically generated**

**Check application working  
A screenshot of a computer

Description automatically generated**

**Update in welcome application which will run on app service.**

**A screenshot of a computer

Description automatically generated**

**Tested locally run works good. Let’s deploy on app service.**

**Build the app and deploy.**

* 1. **Build and create image for the app**
  2. **Push to dockerhub and use that image  
       
     docker file-**

**# Use the official Node.js image as the base image**

**FROM node:14**

**# Set the working directory inside the container**

**WORKDIR /app**

**# Copy package.json and package-lock.json to the working directory**

**COPY package\*.json ./**

**# Install the dependencies**

**RUN npm install**

**# Copy the rest of the application files to the working directory**

**COPY . .**

**# Build the Angular application**

**RUN npm run build**

**# Use the official Nginx image to serve the Angular application**

**FROM nginx:alpine**

**# Copy the build output to the Nginx HTML directory**

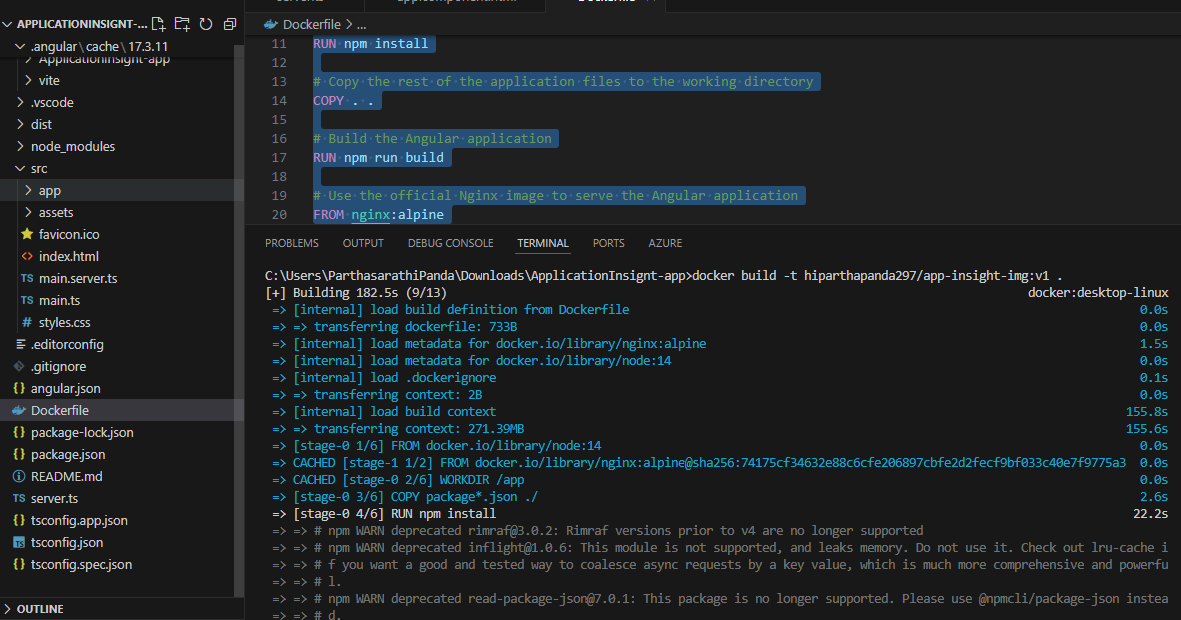
**COPY --from=0 /app/dist/your-angular-app /usr/share/nginx/html**

**# Expose port 80**

**EXPOSE 80**

**# Start Nginx**

**CMD ["nginx", "-g", "daemon off;"]**

**Location – in project directory  
 **

**Error (Backlog) –**

**Could not setup with angular application.**

**Next – setup with react app**

**Bavklog- failed to implement. Deployment failed.**