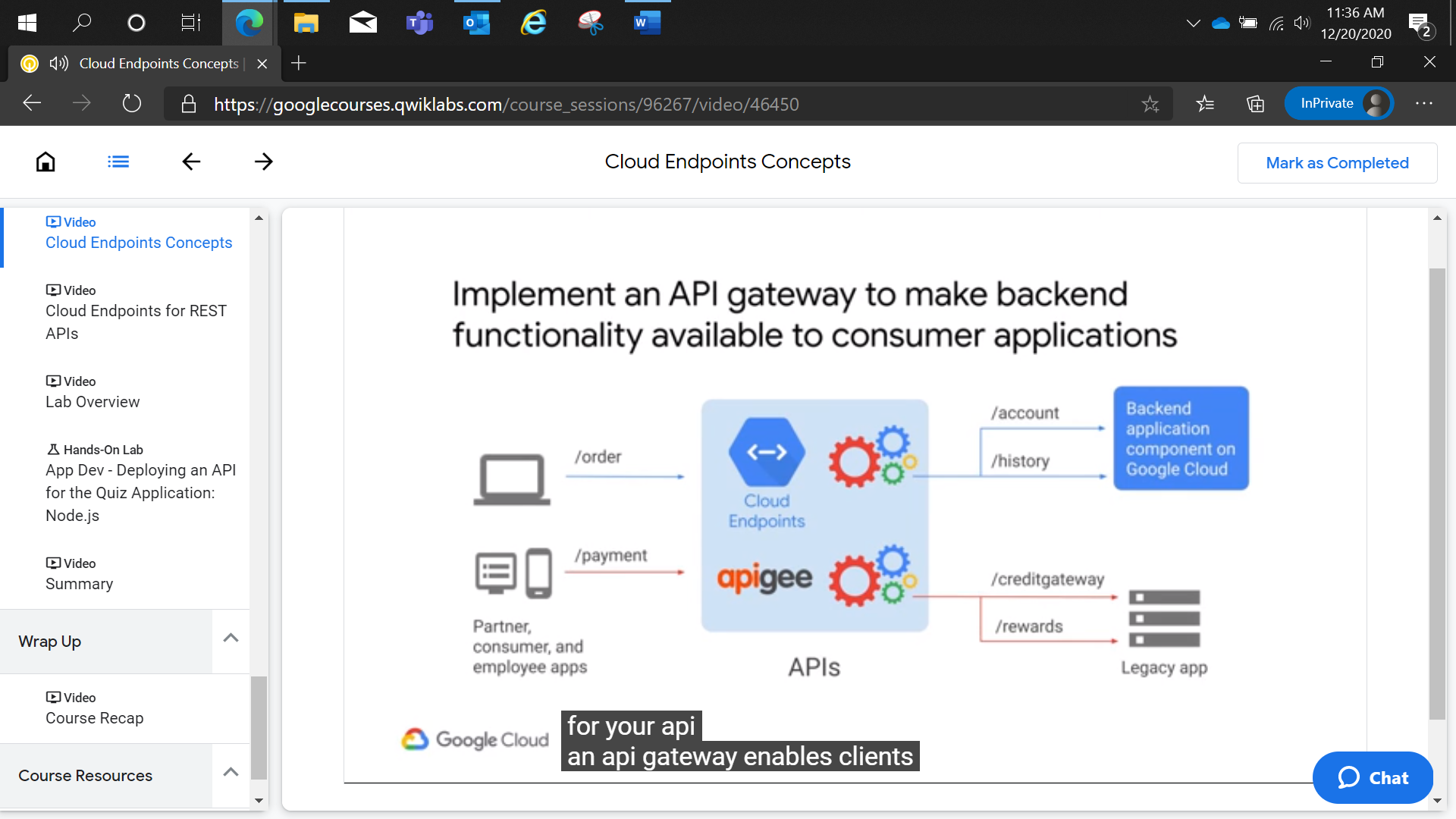
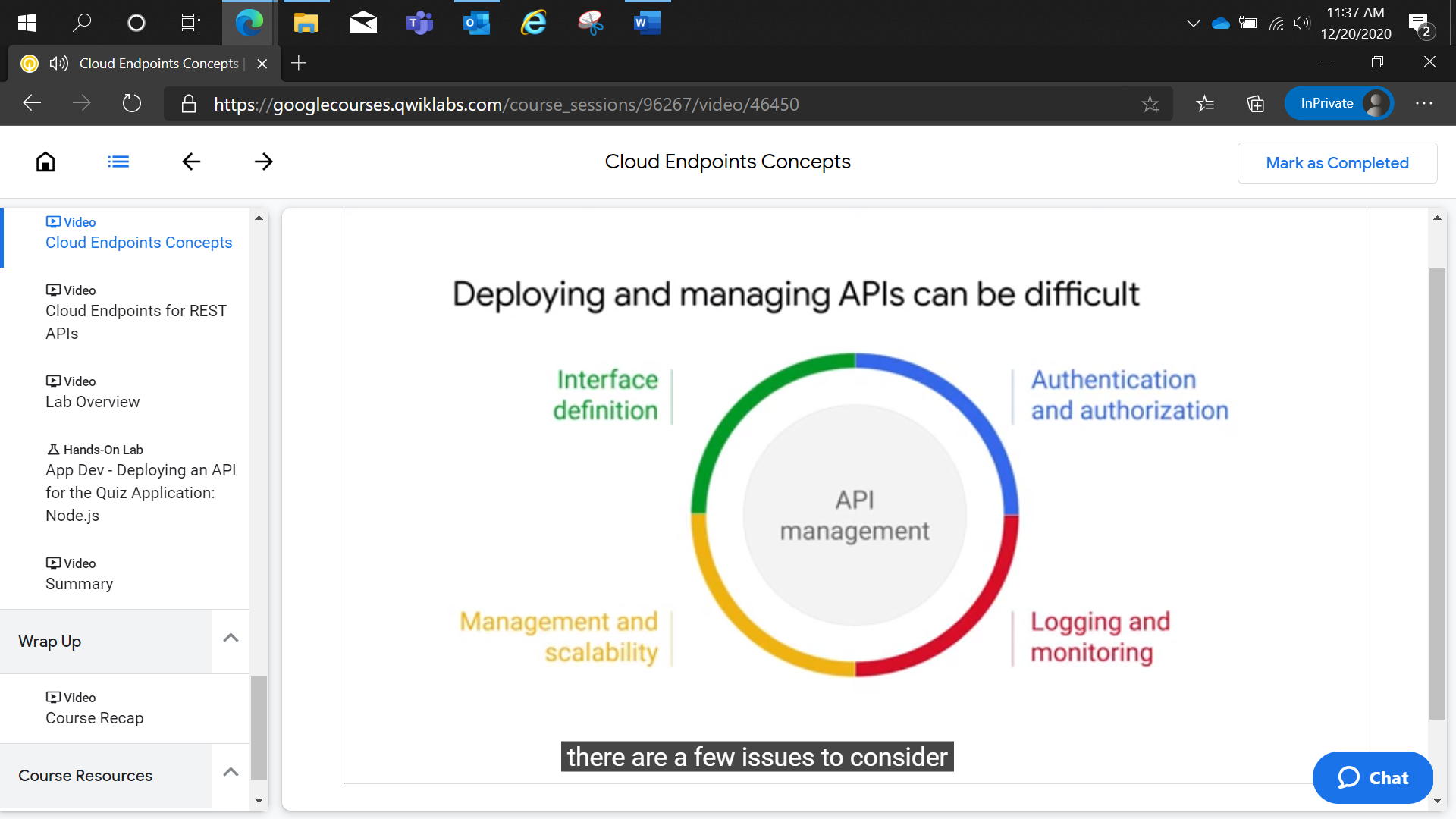
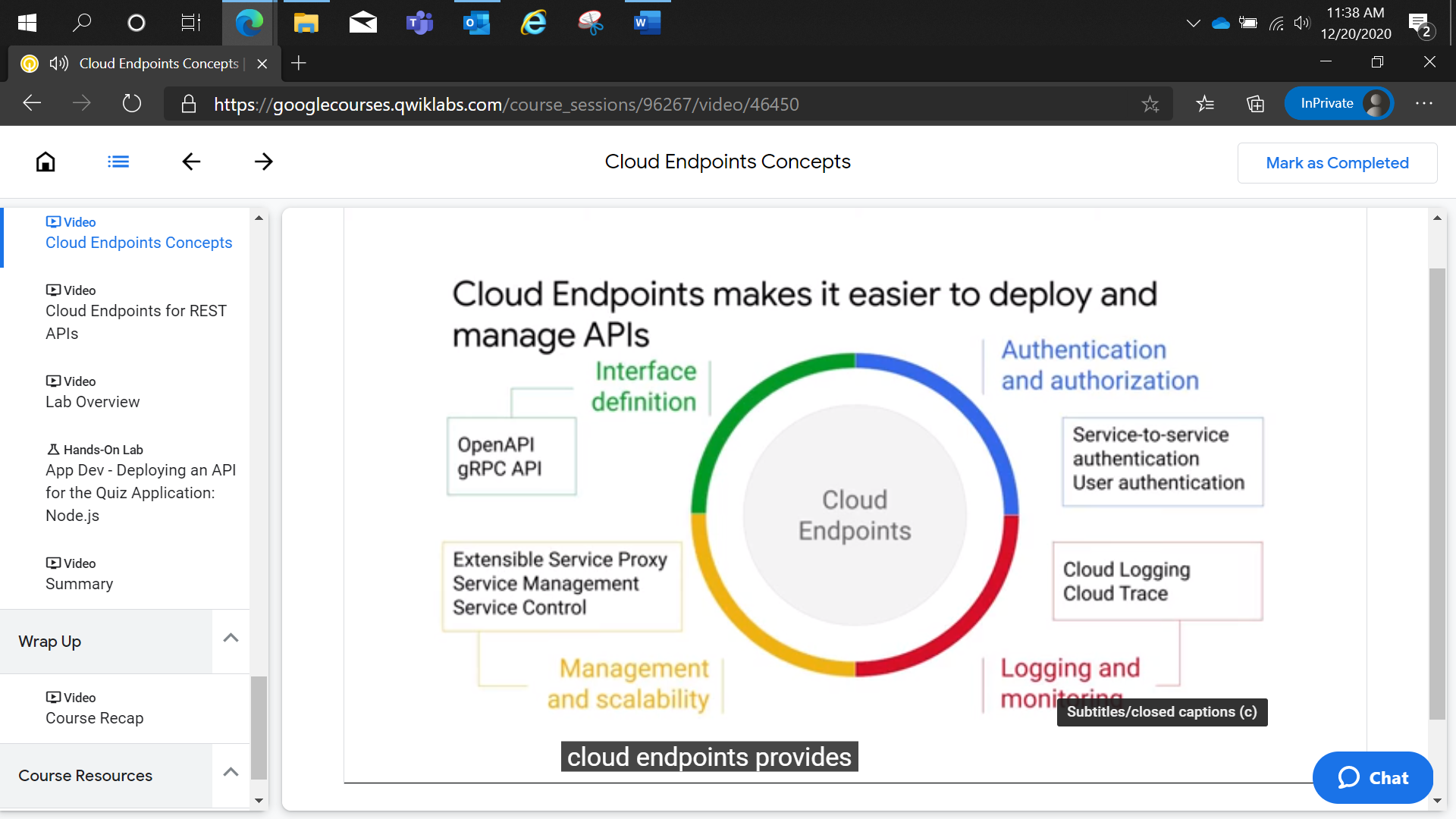
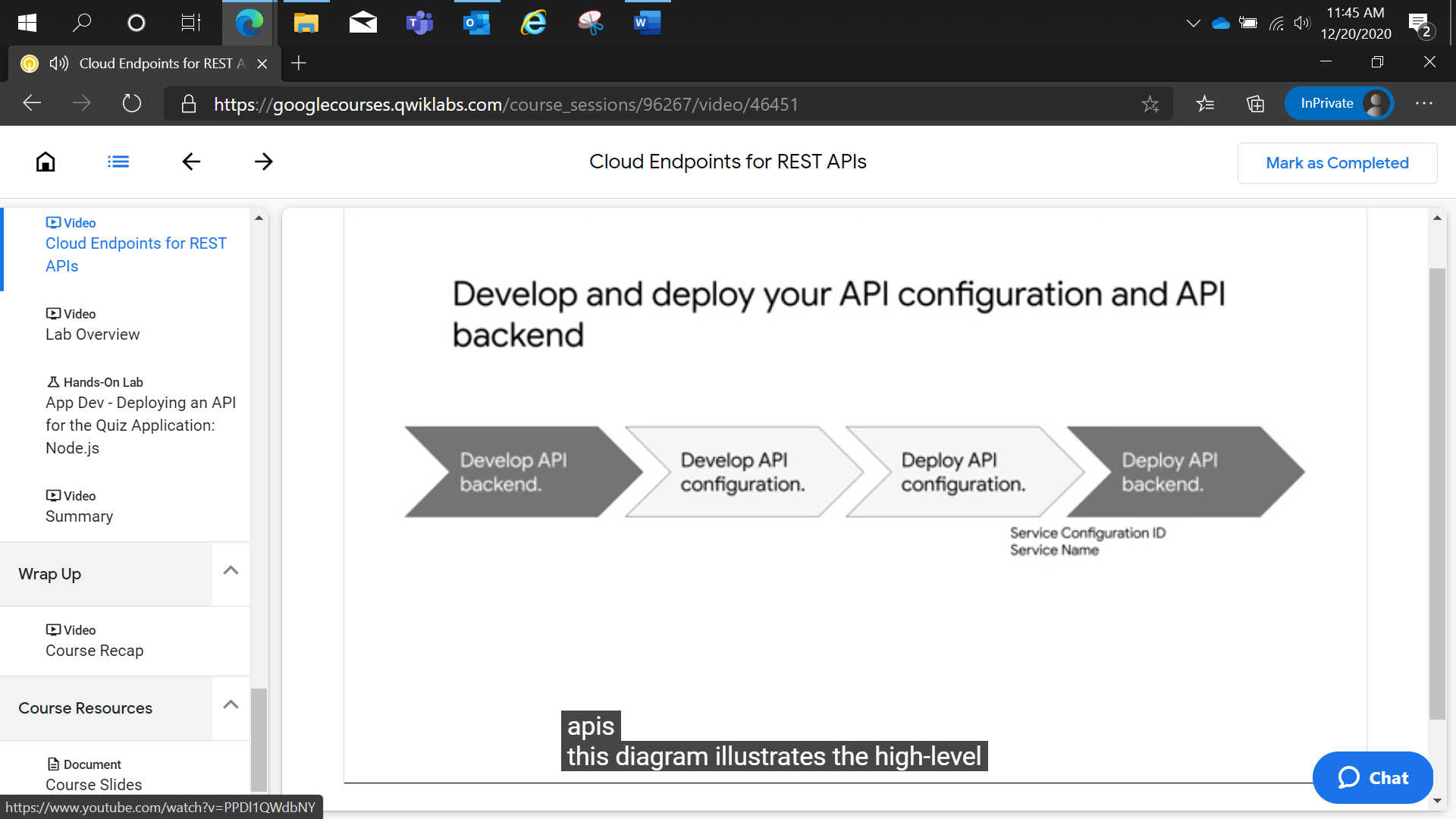
Chapter 1.3.4 Managing APIs with Cloud End points

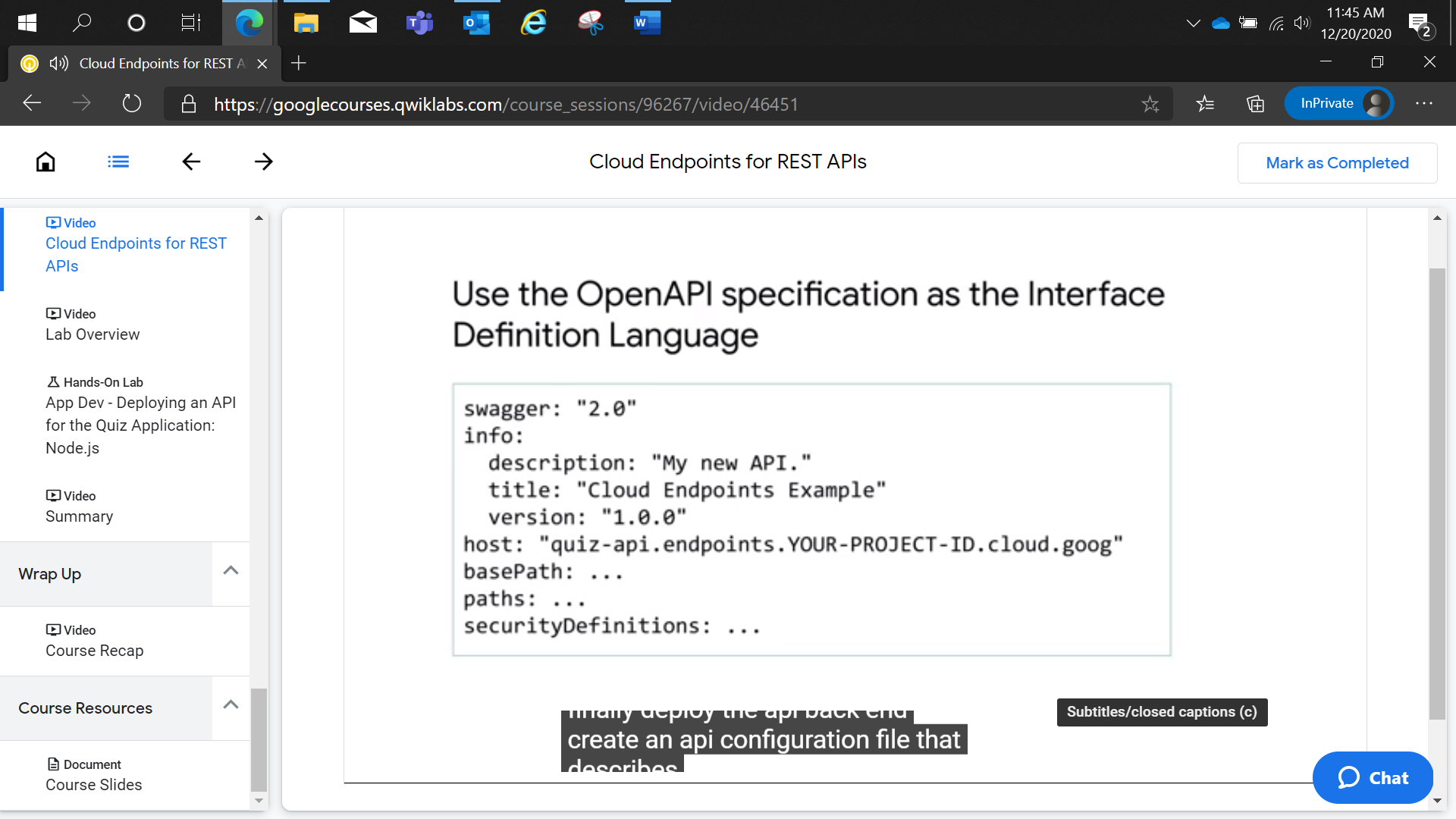


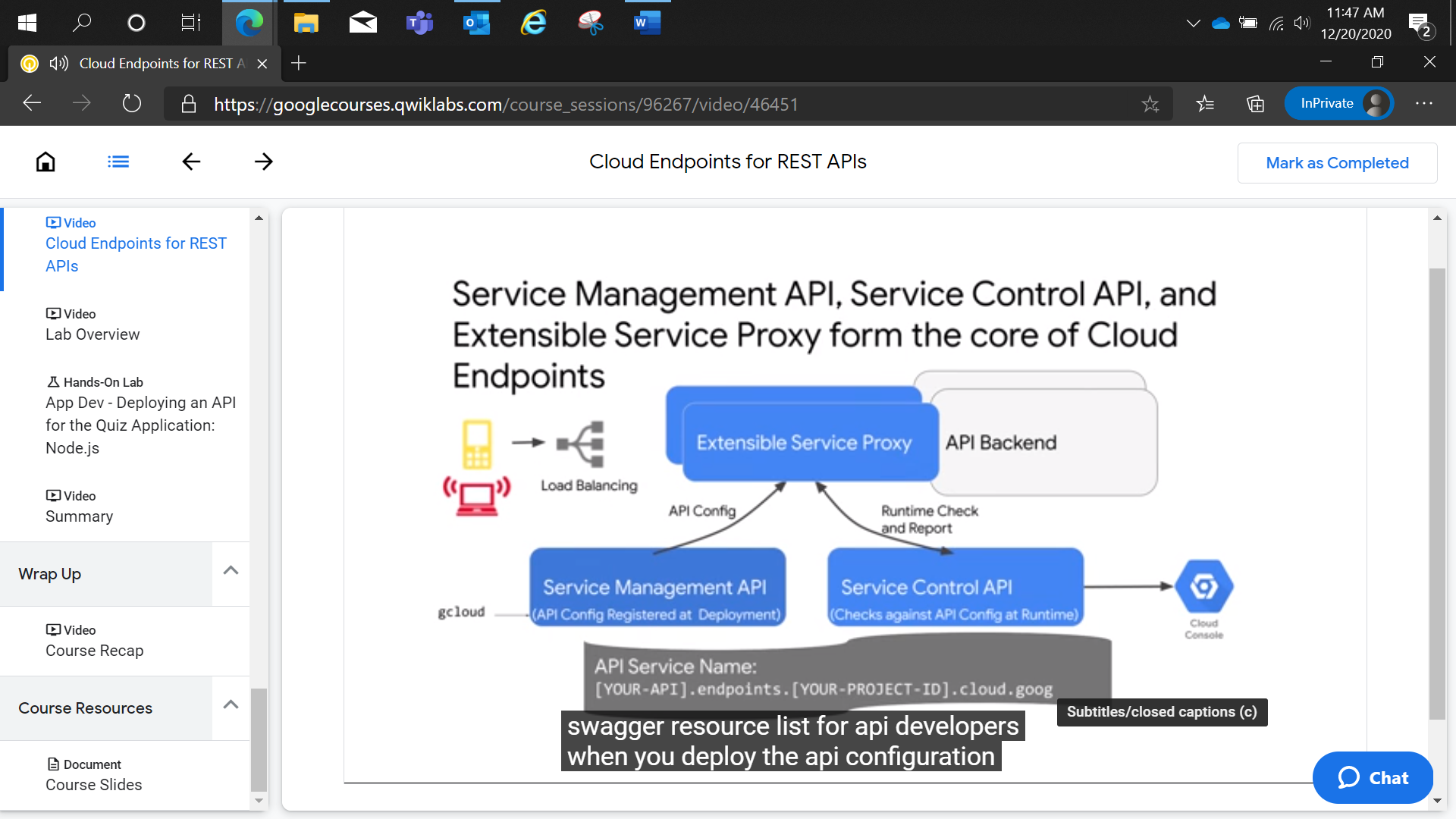


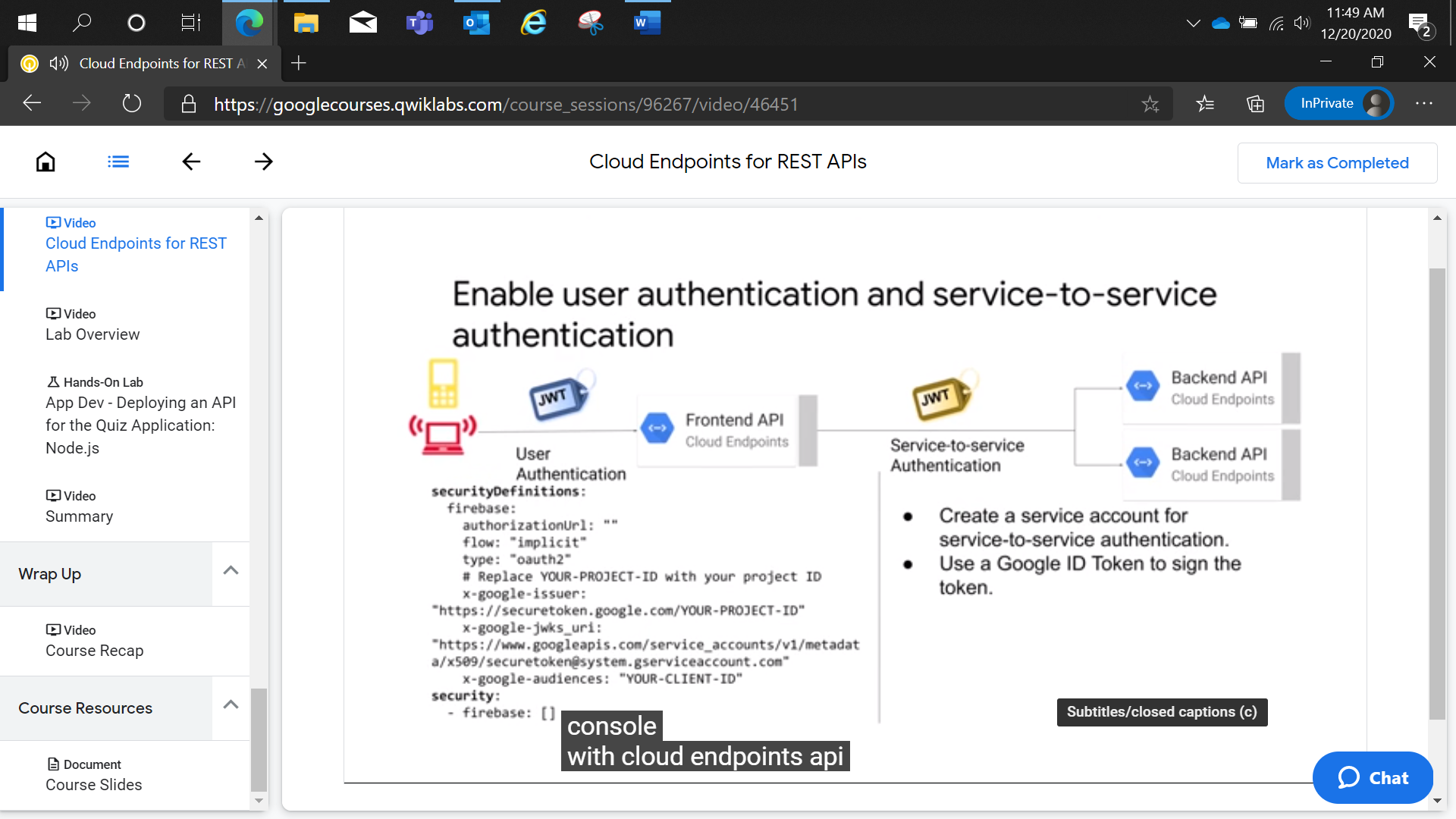


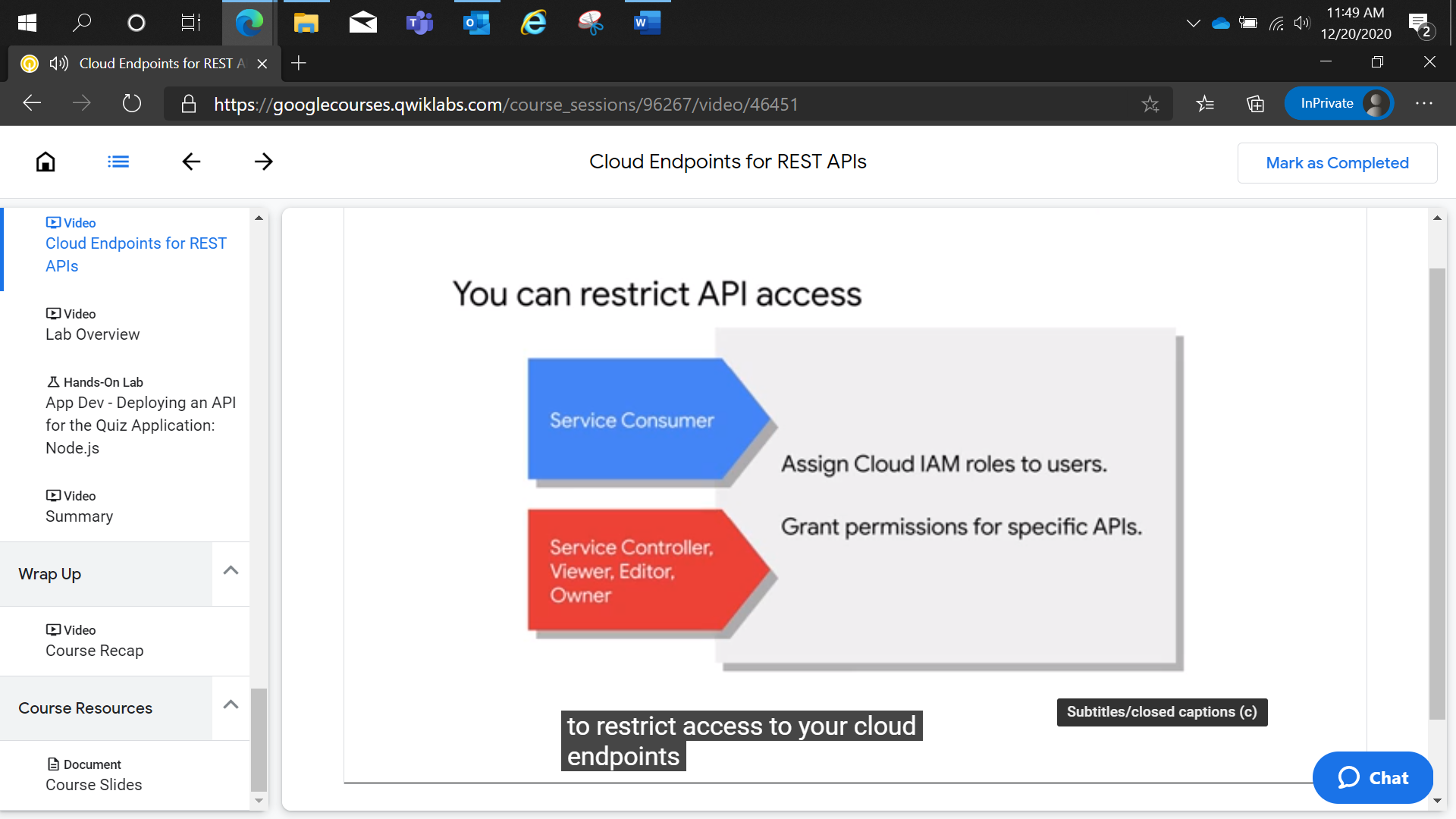


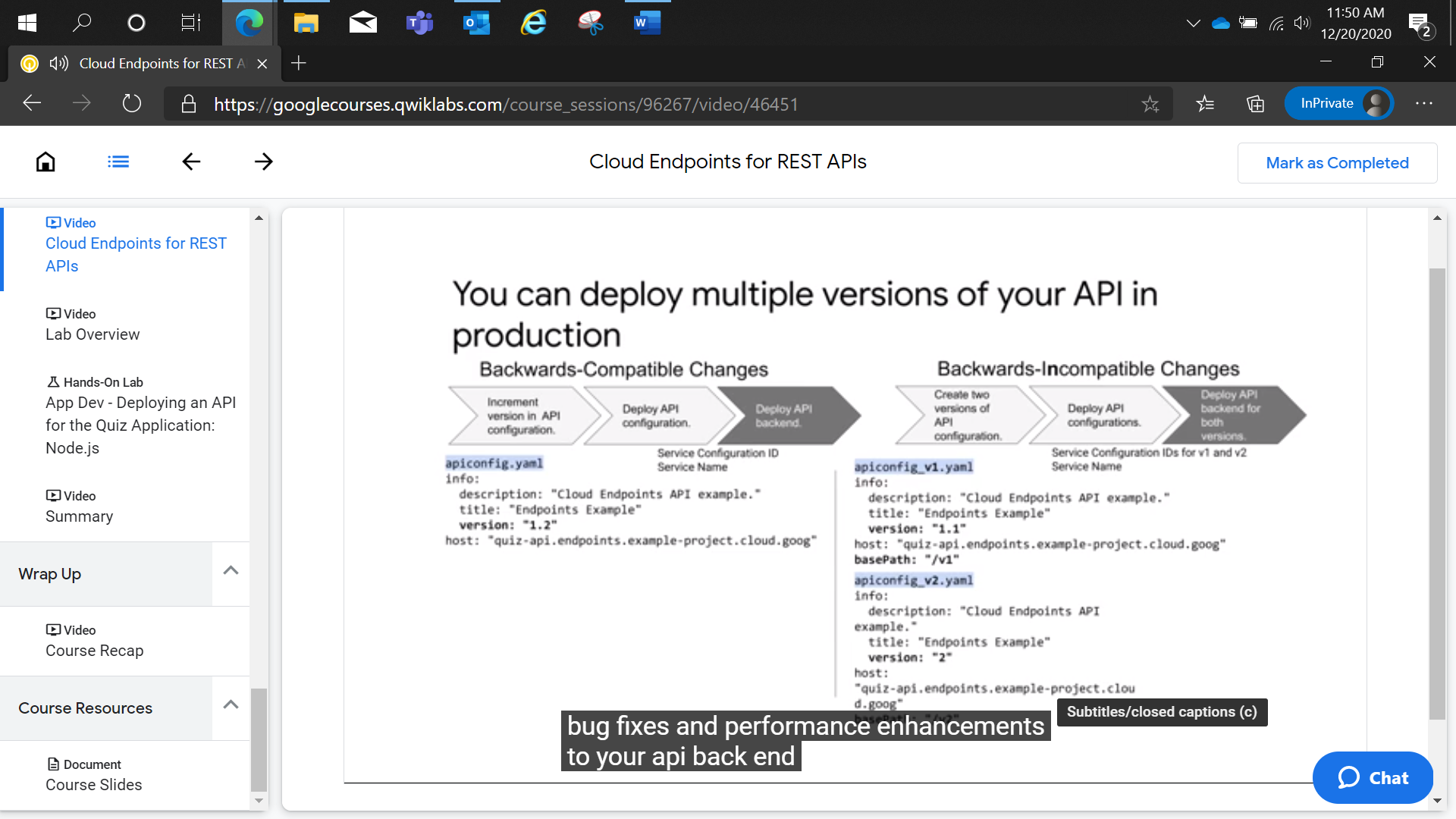


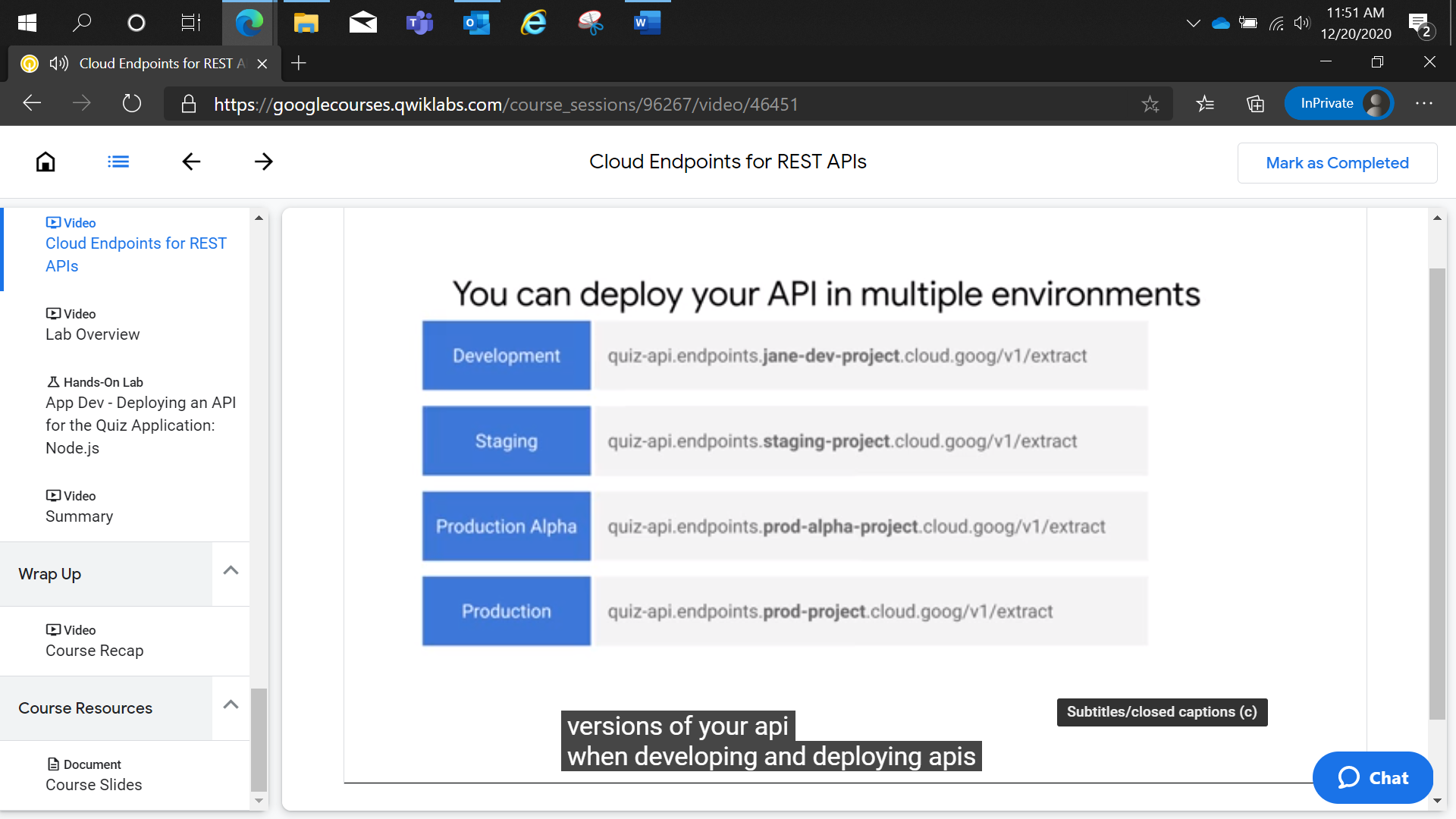


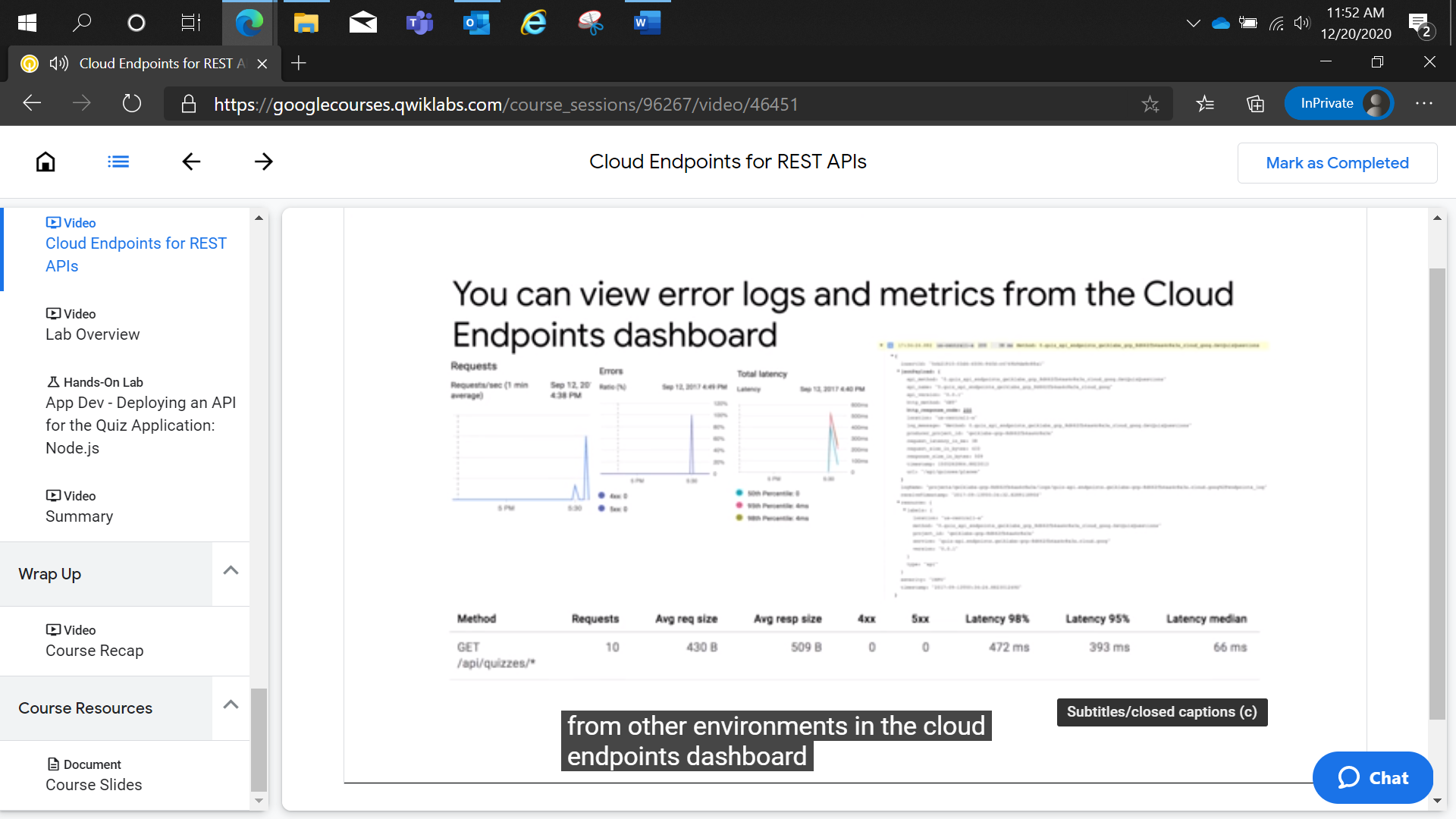


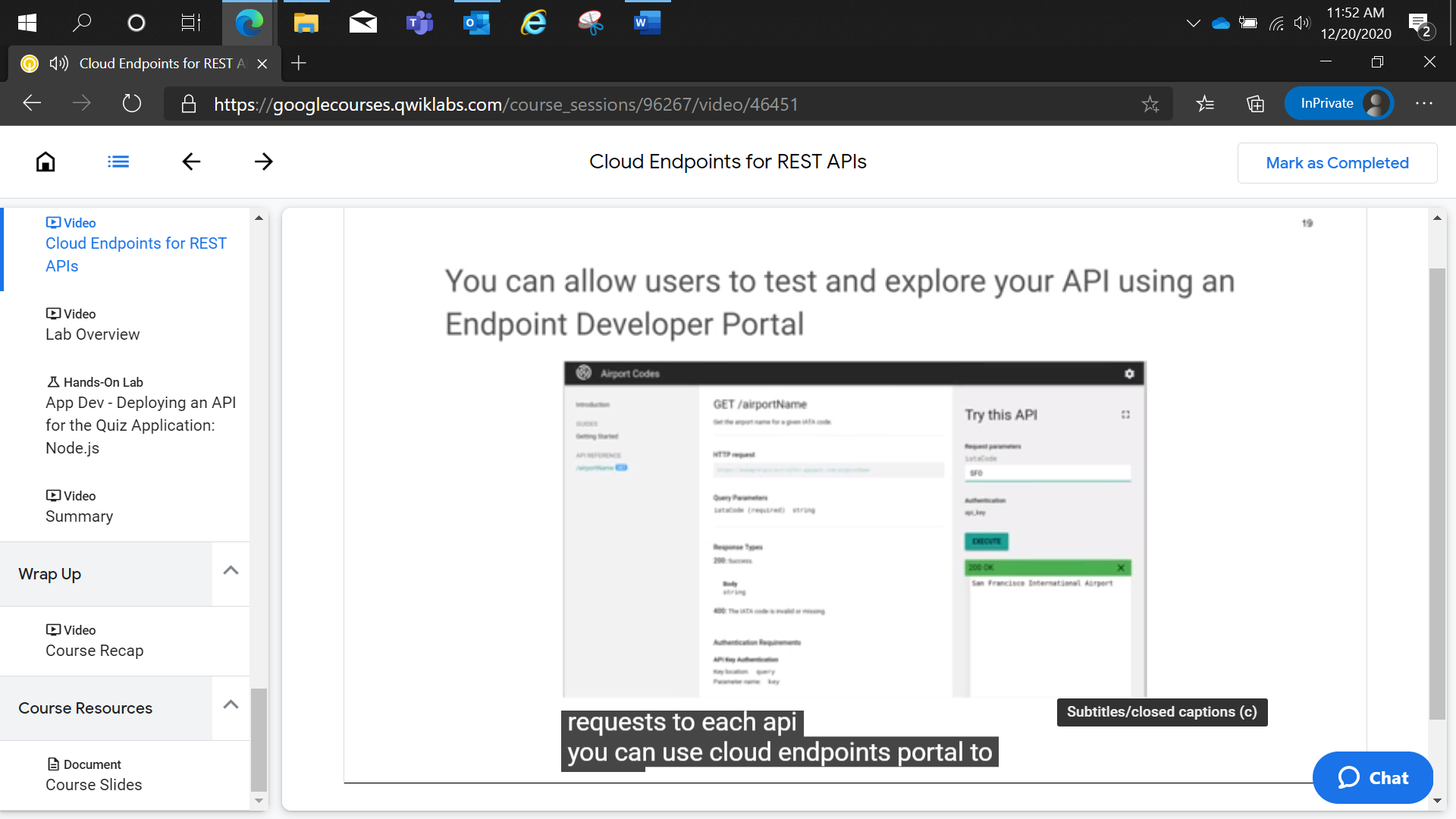


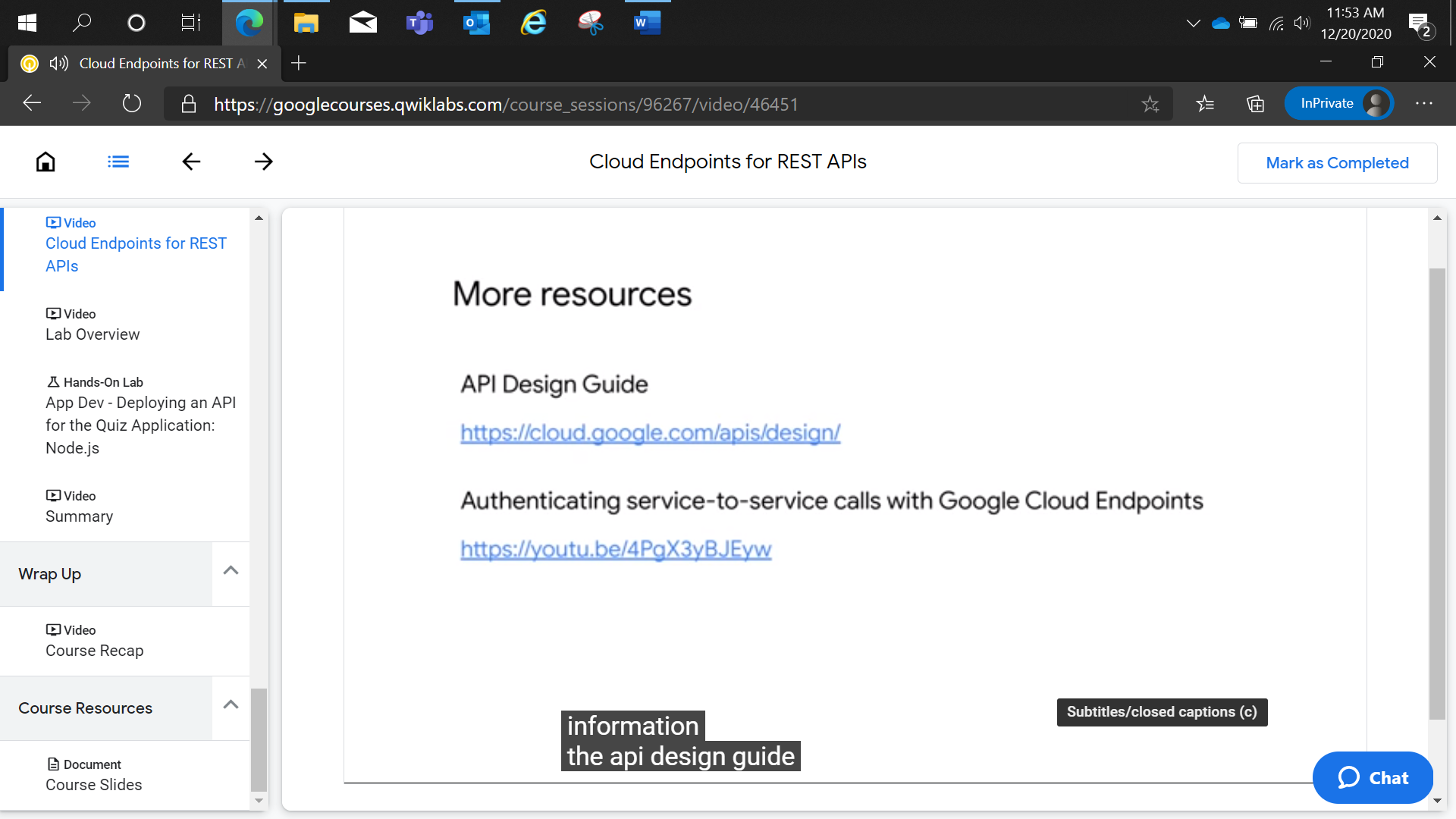














App Dev - Deploying an API for the Quiz Application: Node.js

2 hoursFree

Rate Lab

**Overview**

In this lab, you deploy the Quiz application API into Google Compute Engine and leverage Cloud Endpoints to provide monitoring functionality.

**Objectives**

In this lab, you learn how to perform the following tasks:

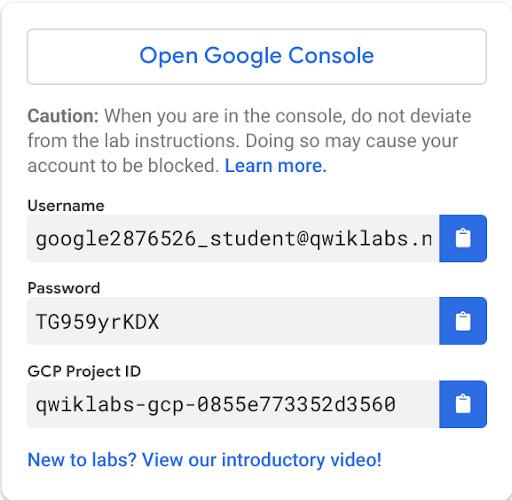
* Create an Open API specification from the existing Quiz application REST API.
* Deploy the specification as a Cloud Endpoint.
* Provision a Compute Engine instance with the Extensible Service Proxy to host the Cloud Endpoints API.

**Setup and requirements**

For each lab, you get a new GCP project and set of resources for a fixed time at no cost.

1. Make sure you signed into Qwiklabs using an **incognito window**.
2. Note the lab's access time (for example,  and make sure you can finish in that time block.

There is no pause feature. You can restart if needed, but you have to start at the beginning.

1. When ready, click .
2. Note your lab credentials. You will use them to sign in to Cloud Platform Console. 
3. Click **Open Google Console**.
4. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.

If you use other credentials, you'll get errors or **incur charges**.

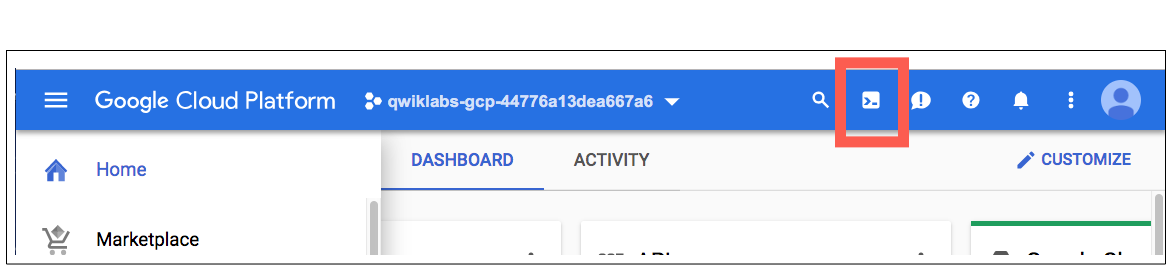
1. Accept the terms and skip the recovery resource page.

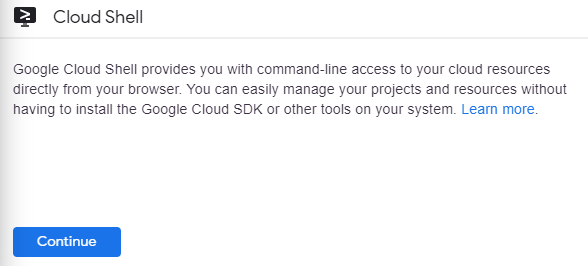
Do not click **End Lab** unless you are finished with the lab or want to restart it. This clears your work and removes the project.

Activate Google Cloud Shell

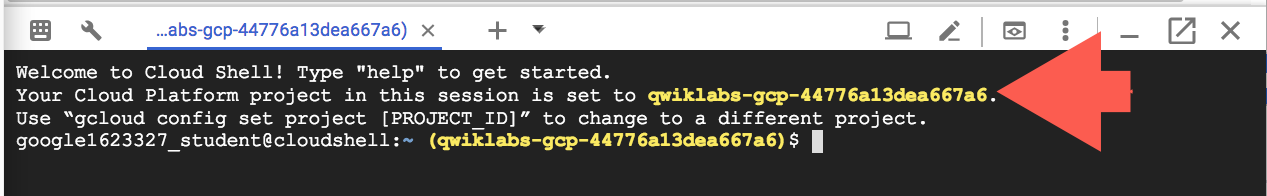
Google Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud. Google Cloud Shell provides command-line access to your GCP resources.

1. In GCP console, on the top right toolbar, click the Open Cloud Shell button.



1. Click **Continue**. 

It takes a few moments to provision and connect to the environment. When you are connected, you are already authenticated, and the project is set to your *PROJECT\_ID*. For example:



**gcloud** is the command-line tool for Google Cloud Platform. It comes pre-installed on Cloud Shell and supports tab-completion.

You can list the active account name with this command:

gcloud auth list

content\_copy

Output:

Credentialed accounts:

- <myaccount>@<mydomain>.com (active)content\_copy

Example output:

Credentialed accounts:

- google1623327\_student@qwiklabs.netcontent\_copy

You can list the project ID with this command:

gcloud config list project

content\_copy

Output:

[core]

project = <project\_ID>content\_copy

Example output:

[core]

project = qwiklabs-gcp-44776a13dea667a6content\_copy

Full documentation of **gcloud** is available on [Google Cloud gcloud Overview](https://cloud.google.com/sdk/gcloud).

**Preparing the Case Study Application**

In this section, you access Cloud Shell, clone the git repository containing the Quiz application, configure environment variables, and run the application.

**Clone source code in Cloud Shell**

1. To clone the repository for the class, execute the following command:

git clone https://github.com/GoogleCloudPlatform/training-data-analystcontent\_copy

1. Create a soft link as a shortcut to the working directory:

ln -s ~/training-data-analyst/courses/developingapps/v1.2/nodejs/cloudendpoints ~/cloudendpointscontent\_copy

**Configure and run the case study application**

1. Change to the directory that contains the sample files for this lab.:

cd ~/cloudendpoints/startcontent\_copy

1. To configure the Quiz application, execute the following command:

. prepare\_environment.shcontent\_copy

When prompted, enter Y to allow unauthenticated invocations of new function **[process-feedback]**.

This script file

* + Creates an App Engine application.
  + Exports environment variables GCLOUD\_PROJECT and GCLOUD\_BUCKET.
  + Runs npm install.
  + Creates entities in Cloud Datastore.
  + Installs an Open API generator tool, api2swagger.
  + Prints out the Google Cloud Platform Project ID.

If you see ERROR: (gcloud.functions.deploy), retry the full gcloud functions deploy command you see after running cat ./prepare\_environment.sh.

You can run gcloud functions list to verify the function is created and available.

1. To run the web application, execute the following command:

npm startcontent\_copy

The application is now running from Cloud Shell.

1. In **Cloud Shell**, click **Web preview** > **Preview on port 8080** to preview the quiz application.
2. Modify the URL by adding /api/quizzes/places to the end of the hostname.

Replace ?authuser=0 with /api/quizzes/places.

You should see JSON data returned from the quiz application API corresponding to the questions in the places quiz.

1. Make a note of the complete API URL to use later in this lab.

Save the url so you can paste it in a command in the next section.

The URL will be in the form: https://8080-dot-#######-dot-devshell.appspot.com/api/quizzes/places

Click *Check my progress* to verify the objective.

Configure and run the case study application

Check my progress

**Generating an Open API Specification**

In this section, you will generate an Open API specification that will form the basis for deploying your Cloud Endpoint.

**Create an Open API Specification**

1. Open a second **Cloud Shell** window.

Click Open a new tab.

1. To change the directory to the Cloud Endpoints working folder, execute the following command:

cd ~/cloudendpoints/start/endpointcontent\_copy

1. Update the PATH environment variable to set the path for the api2swagger command:

export PATH=$PATH:`npm root -g`/api2swagger/bincontent\_copy

1. To generate the Open API specification, execute the following command:

api2swagger -e [API\_URL\_FROM\_CLIPBOARD] -o ./quiz-api.jsoncontent\_copy

Replace [API\_URL\_FROM\_CLIPBOARD] with the **API URL** saved in the previous step of form https://8080-dot-.../api/quizzes/places.

The api2swagger tool will make the request against the API and display a series of prompts.

If you don't complete the proper replacement, you'll see an **Invalid hostname** error.

1. Enter responses to the api2swagger prompts using the following table:

|  |  |
| --- | --- |
| **Prompt** | **Response** |
| Title of Swagger Spec ? | Quite Interesting Quiz API |
| Description of Swagger Spec ? | An API for the Quite Interesting Quiz |
| Terms of Service URL | (Press enter key) |
| Version of your API Program ? | (Press enter key) |
| Contact Name? | (Press enter key) |
| Contact URL ? | (Press enter key) |
| Contact Email ? | (Press enter key) |
| License Name ? | (Press enter key) |
| License URL ? | (Press enter key) |
| Does your API support http ? | Yes |
| Pick Base Path from your API ? | Use the up and down arrows to select:  **❯ /api/quizzes**  (Press enter key) |
| A verbose explanation of the operation behavior ? | Gets questions for a quiz |
| A short summary of what the operation does ? | Get quiz questions |
| Additional external documentation for this operation ? | (Press enter key) |
| Unique string used to identify the operation ? | getQuizQuestions |
| A list of tags for API documentation control ? | (Press enter key) |
| API Path has any dynamic parameters ? | Y (Press enter key) |
| Choose Dynamic Params in URL ? | Use the up and down arrows to move to:**❯◉ places**  Then **press the space bar to select places**, then press enter |
| Name of URL Param ? | quizName |
| Description of URL Param ? | The name of the quiz |
| Type of query param ? | Use the up and down arrows to select:**❯ string**  (Press enter key) |

The api2swagger tool will display the output for the Open API specification.

**Edit the Open API Specification**

1. In **Cloud Shell**, click **Open Editor**.

You may need to click **Open in New Window** if third-party cookie blocking is enabled.

1. Navigate to **cloudendpoints/start/endpoint**.
2. In the **Cloud Shell Code Editor**, open quiz-api.json.
3. Replace the value for the "host" key with a hostname using a string in the form quiz-api.endpoints.[Project-ID].cloud.goog.

The following sample shows the modification in the **host** line.

Be sure you replace **[GCP-Project-ID]** with your GCP Project ID.

You can find the GCP Project ID in the left panel of the lab instructions under your credentials.

**endpoint/quiz-api.json**

{

"swagger": "2.0",

"host": "quiz-api.endpoints.qwiklabs-gcp-XX-XXXX.cloud.goog",

"schemes": [

"https",

"http"

],content\_copy

When you deploy your Open API specification in the next section, the Service Management API will use the host value in your deployment configuration file to create a new Cloud Endpoints service with the name quiz-api.endpoints.[PROJECT-ID].cloud.goog (if it does not exist), and then configure the service according to your OpenAPI configuration file. Cloud Endpoints uses DNS-compatible names to uniquely identify services. Because projects in Google Cloud Platform are guaranteed to have a globally unique name, you can use your project name to create a unique API service name. You can also map your own DNS name to your API.

1. Save the file.

**Deploy the API Specification to Cloud Endpoints**

1. To deploy the Open API specification as a Cloud Endpoint, execute the following command in the **Cloud Shell** window:

Click **Open Terminal** to return to Cloud Shell.

gcloud endpoints services deploy quiz-api.jsoncontent\_copy

If you see an error like PERMISSION\_DENIED: Ownership for domain name, make sure your **host** in the quiz-api.json file matches the suggested format including the correct your project-id.

The service definition takes a few minutes to deploy.

You will see a warning related to the API being available to all clients without supplying a key.

1. To view the Open API configuration name, execute the following command:

gcloud endpoints configs list --service="quiz-api.endpoints.$GOOGLE\_CLOUD\_PROJECT.cloud.goog"content\_copy

Notice the project-id has been inserted. You can copy the **service** from the output of your deploy command.

The service configuration will be returned, including the **CONFIG\_ID** and **SERVICE\_NAME**.

You will need the both values in the next section.

**Deploying the API Backend**

In this section, you will provision a Compute Engine instance to run the API implementation and the Cloud Endpoints Extensible Service Proxy.

**Create a Compute Engine Instance**

1. In the **Cloud Platform Console**, on the **Navigation menu**, click **Compute Engine**.
2. On the **VM Instances** page, click **Create**.
3. On the **Create an instance** page, specify the following:

|  |  |
| --- | --- |
| **Property** | **Value** |
| Name | endpoint-host |
| Region | us-central1 |
| Zone | us-central1-a |
| Boot disk | Make sure **Debian GNU** is selected |
| Identity & API access > Access scopes | **Allow full access to all Cloud APIs** |
| Firewall | **Allow HTTP traffic** |
| Management, security, disks, networking, sole tenancy > Automation > Metadata | Add the following keys and values |
| **Metadata Key** | **Value** |
| endpoints-service-config-id | The CONFIG\_ID for the endpoint in the form:yyyy-mm-ddr0 |
| endpoints-service-name | The SERVICE\_NAME for the endpoint in the form:  quiz-api.endpoints.<Project-ID>.cloud.goog |

1. Click **Create**.

The Compute Engine instance will take around 20 seconds to provision with Debian.

Install and run the API Backend

1. Return to the **Cloud Shell** window.
2. To ensure that the working directory is set to the endpoint folder, execute the following command:

cd ~/cloudendpoints/start/endpoint/content\_copy

1. To copy the application source files from **Cloud Shell** to **endpoint-host**, execute the following command:

gcloud compute scp ./quiz-api endpoint-host:~/ --recurse --zone=us-central1-acontent\_copy

You may be prompted to create an SSH key. You can press the ENTER key for each prompt including the Y to create the .ssh directory.

It may take a few seconds to copy the files.

1. Return to the **Cloud Platform Console**.
2. On the **Compute Engine** page, to connect to the **endpoint-host** virtual machine, click **SSH**.

You may need to click twice if you have a popup blocker.

It will take a few seconds to establish the SSH connection.

1. In the **endpoint-host** SSH window, install the software prerequisites:

curl -sL https://deb.nodesource.com/setup\_12.x | sudo -E bash -content\_copy

sudo apt-get install -y nodejscontent\_copy

curl -L https://npmjs.org/install.sh | sudo shcontent\_copy

Click *Check my progress* to verify the objective.

Create a Compute Engine Instance and install the software prerequisites

Check my progress

1. To run the quiz API application, execute the following commands within the SSH session:
2. export GCLOUD\_PROJECT="$(curl -H Metadata-Flavor:Google http://metadata/computeMetadata/v1/project/project-id)"
3. export GCLOUD\_BUCKET=$GCLOUD\_PROJECT-media
4. export PORT=8081
5. cd ~/quiz-api
6. npm install

npm startcontent\_copy

1. Return to the **Cloud Platform Console**, which should still display the **Compute Engine** VM instance list.
2. Establish a second **SSH** connection to **endpoint-host**.
3. In the second **endpoint-host** SSH window, to install the Cloud Endpoints Extensible Service Proxy by executing the following commands:
4. export CLOUD\_ENDPOINTS\_REPO="google-cloud-endpoints-jessie"
5. echo "deb http://packages.cloud.google.com/apt $CLOUD\_ENDPOINTS\_REPO main" | sudo tee /etc/apt/sources.list.d/google-cloud-endpoints.list
6. curl --silent https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -

sudo apt-get update && sudo apt-get install google-cloud-sdk && sudo apt-get install endpoints-runtimecontent\_copy

1. Still in the second **endpoint-host** SSH window, execute the following command to edit the nginx configuration file:

sudo nano /etc/default/nginxcontent\_copy

1. To modify the file, add the following declaration after the existing configuration:
2. PORT=80

content\_copy

1. Save the file (press **Ctrl+O** then **Enter**) and exit nano (press **Ctrl+X**).
2. To restart nginx, execute the following command:

sudo service nginx restartcontent\_copy

**Testing and Modifying Cloud Endpoint**

In this section, you will verify the deployment of the endpoint and modify the configuration to allow serving via a hostname instead of an IP address.

**Invoke the Cloud Endpoint via the IP address**

1. Return to the **Cloud Platform Console** > **Compute Engine**.
2. To launch a browser window, click on the **endpoint-host** virtual machine's External IP address.

A JSON message reporting **Method does not exist** will be displayed in the new browser window. This is expected behavior, because you have not associated a method with "/". You do not need to take corrective action.

1. Add /api/quizzes/places to the end of the URL.

You should see JSON data returned from the quiz API.

1. In the **Cloud Platform Console**, on the **Navigation menu**, click **Endpoints**, and then click on your deployed endpoint.

You should see a monitoring page for the quiz-api endpoint.

After a few minutes, you should see the request that you made against the endpoint.

**Modify and redeploy the Cloud Endpoint configuration**

1. Return to the **Cloud Shell** code editor, and select the quiz-api.json file.
2. To allow the API to be accessed via the Cloud Endpoint hostname instead of the IP address, add the "x-google-endpoints" key and value shown in the following file fragment:

**endpoint/quiz-api.json**

{

"swagger": "2.0",

"host": "quiz-api.endpoints.<Project-ID>.cloud.goog",

"x-google-endpoints": [ {

"name": "quiz-api.endpoints.<Project-ID>.cloud.goog",

"target": "<endpoint-host-EXTERNAL-IP-ADDRESS>" } ],

"schemes": [

"https",

"http"

],content\_copy

1. Change the values of <Project-ID>, and <endpoint-host-EXTERNAL-IP-ADDRESS> then save the file.
2. To redeploy the API, return to the **Cloud Shell** window, and execute the following command:

gcloud endpoints services deploy quiz-api.jsoncontent\_copy

1. Open a new browser tab, and navigate to: http://quiz-api.endpoints.<Project-ID>.cloud.goog/api/quizzes/gcp

You should see JSON data from the GCP quiz.

**Monitoring the API**

In this section you will review the monitoring output from your API.

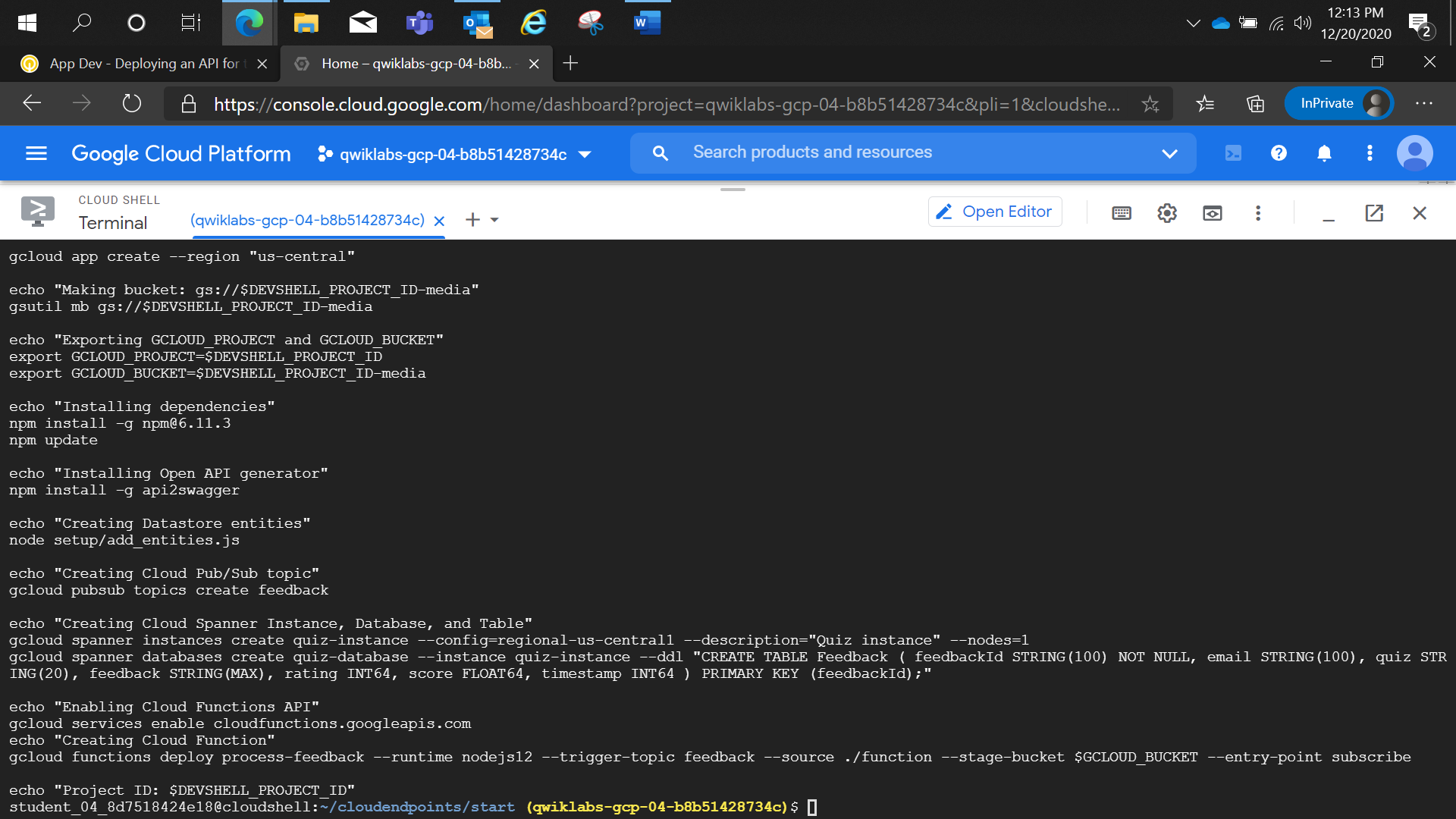
**Inspect the API in the Console**

1. Return to the **Cloud Platform Console**, and on the **Navigation menu**, click **Endpoints**, and then click on your deployed endpoint.

You should see an overview page for the Quiz API.

1. Take a few minutes to review the Requests, Latency, and Error graphs.

You should see that the monitoring graphs allow you to observe the performance of your API as it executes requests from clients.



CONFIG\_ID SERVICE\_NAME

2020-12-20r0 quiz-api.endpoints.qwiklabs-gcp-04-b8b51428734c.cloud.goog

"x-google-endpoints": [ {

"name": "quiz-api.endpoints.qwiklabs-gcp-04-b8b51428734c.cloud.goog",

"target": "35.239.61.179" } ],

http://quiz-api.endpoints.qwiklabs-gcp-04-b8b51428734c.cloud.goog/api/quizzes/gcp