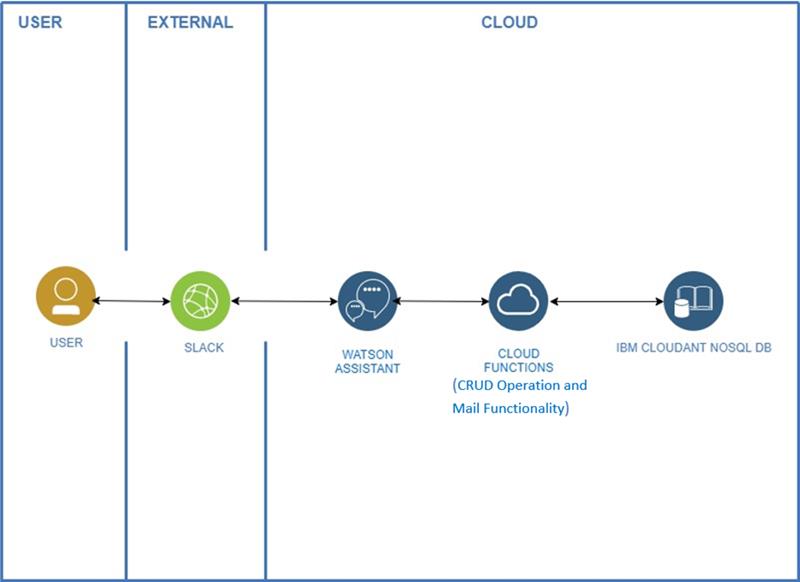
**Idea Summary:**

* In a pandemic situation, the staff availability for handling booking and cancellation of medical appointments is limited and there are more requests. There is a need for an interactive and reliable automated booking and cancellation facility to address the panic from the patients’ end.

**Solution:**

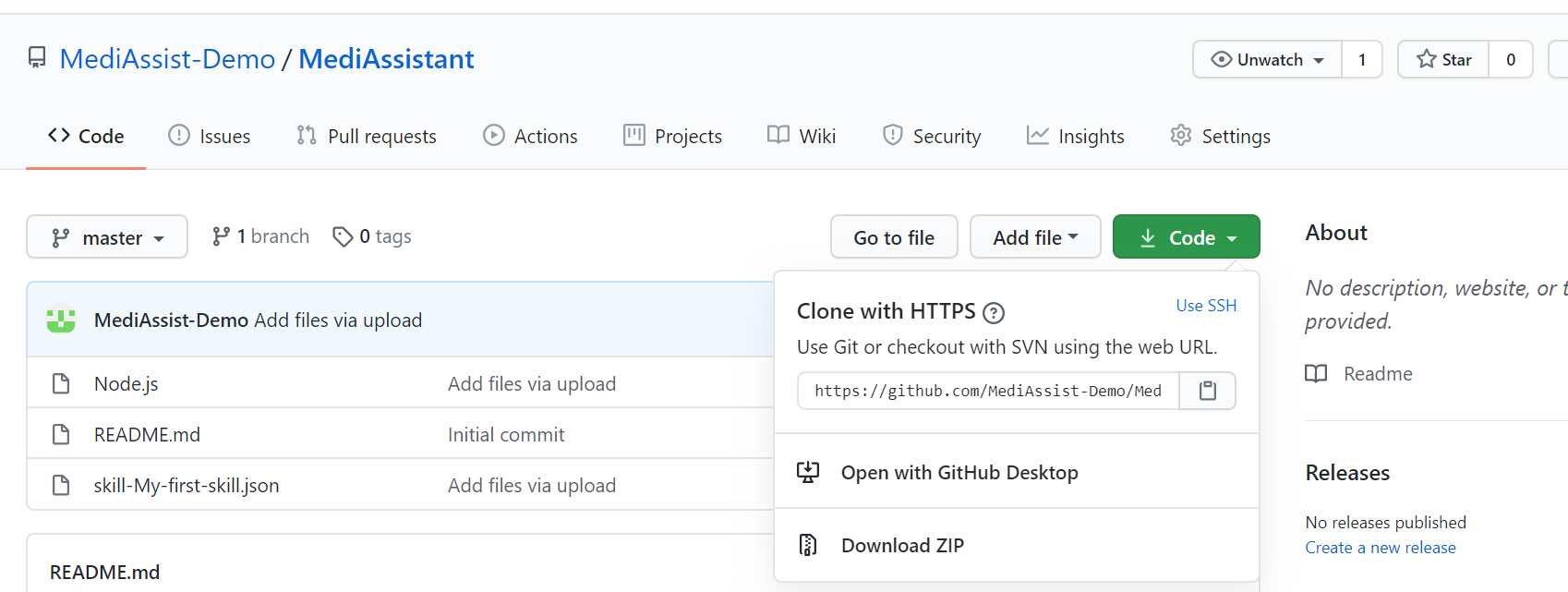
* To address this challenge, We have come up with a Medi Assistant App, a Slack Bot that understands users' requests to book/cancel an appointment and automates the process without the help of staff to perform the process.
* The Medi Assistant app utilizes the IBM Watson Assistant service to collect the necessary details from the user to Book/ Cancel a medical appointment.
* Once the user confirms/cancels the appointment, the details are inserted (or) deleted in the IBM Cloudant database. Also, an Email notification is sent to the user. Have used the IBM Cloud function as a webhook with Watson Assistant to implement the above functionalities.

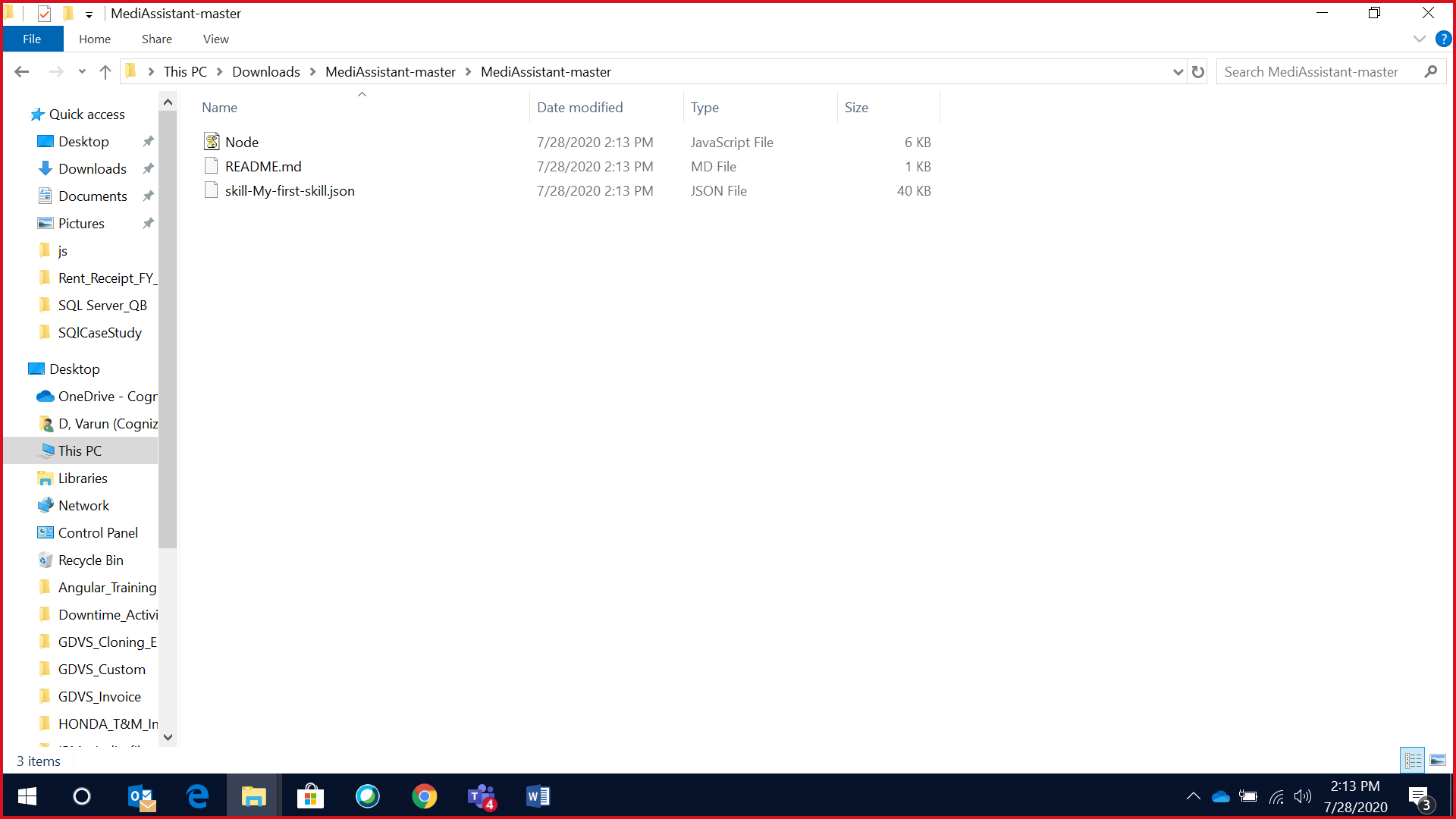
**Architecture Diagram:**



**Step 1:**

* Navigate to theMediAssistantrepository(https://github.com/MediAssistDemo/MediAssistant.git) and download the ZIP folder. Extract the ZIP folder which will provide the files that we have used in this prototype.

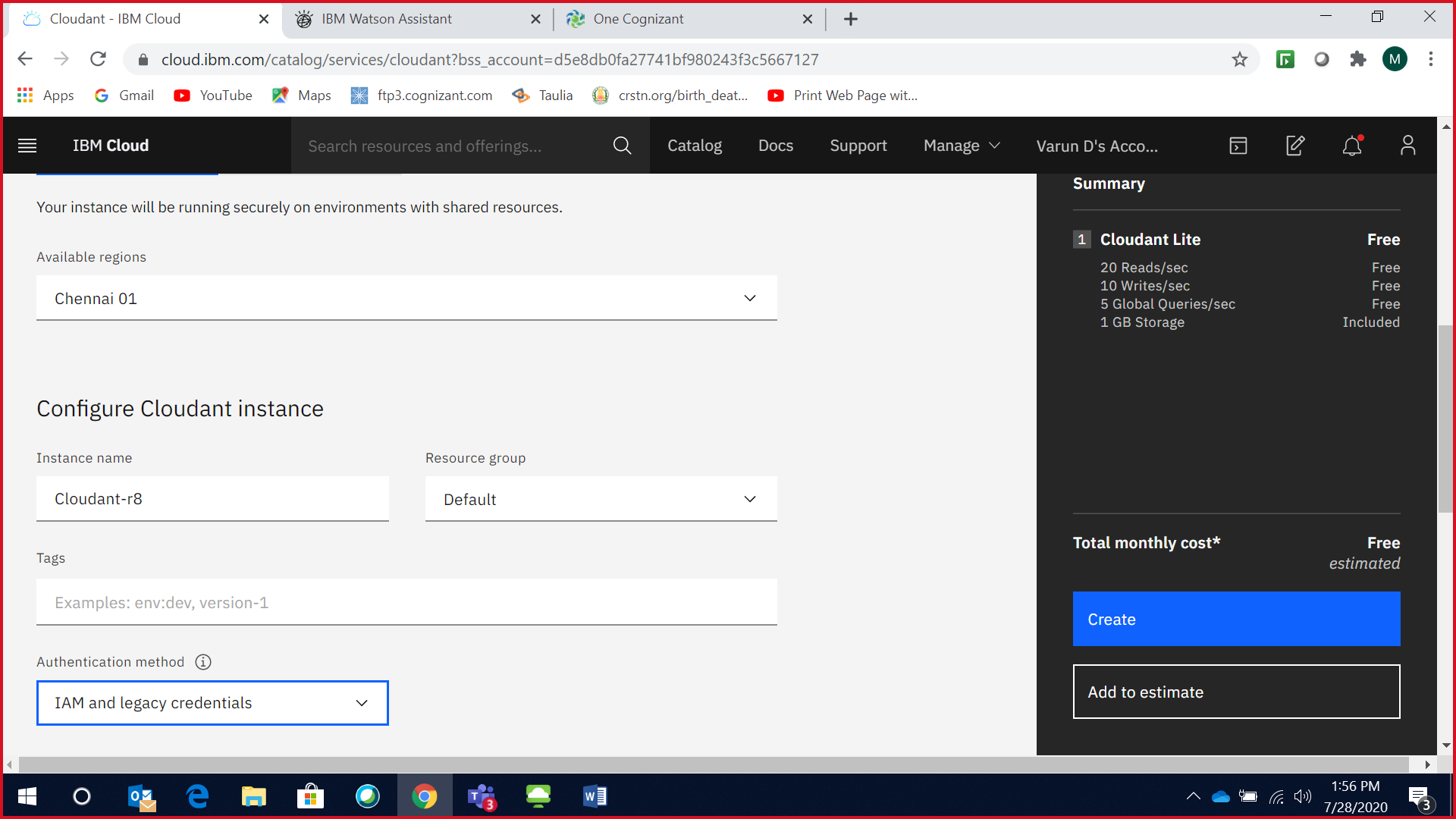




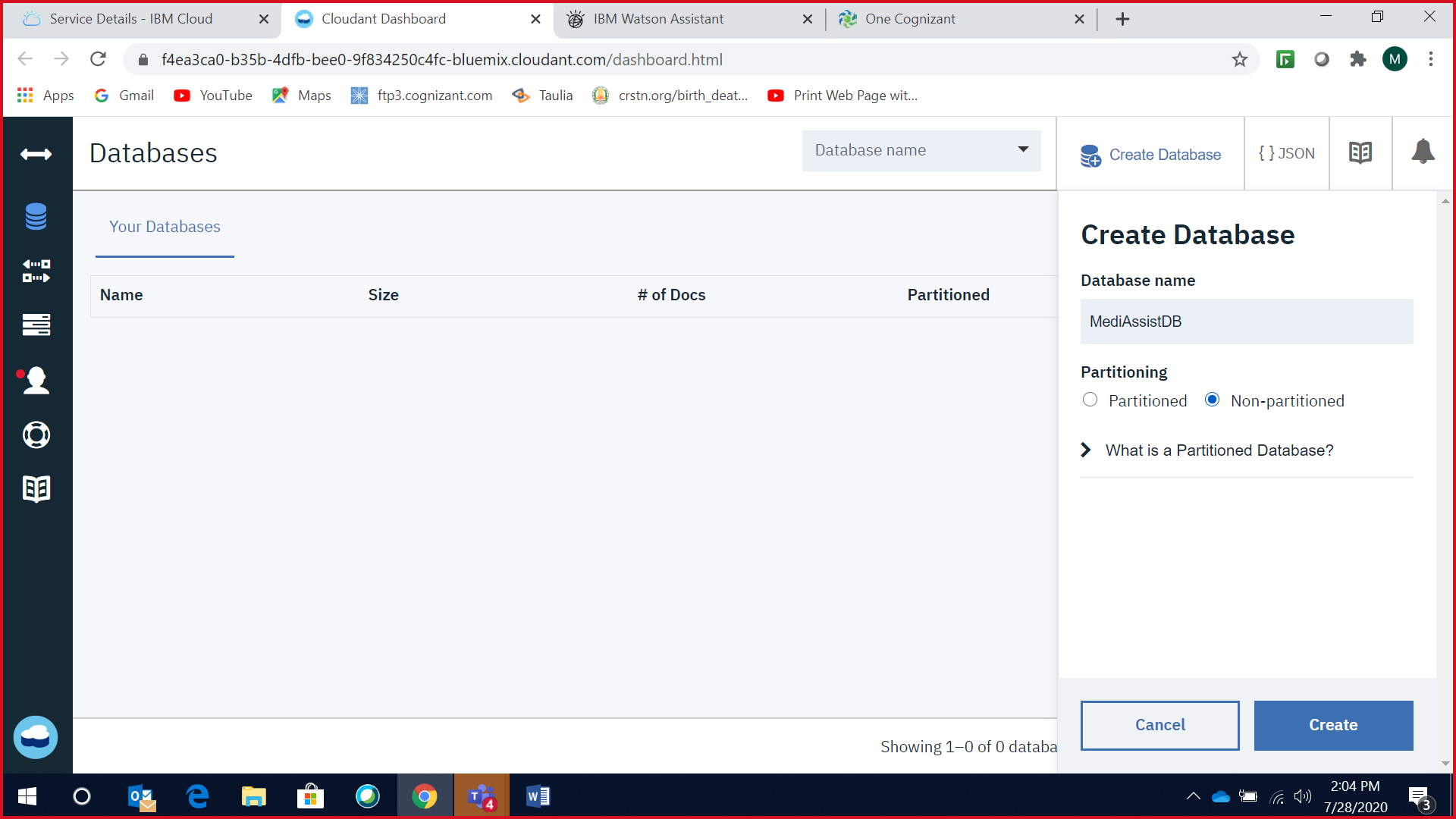
**Step 2:**

* Create a Cloudant service, while creating the Cloudant service make sure to select a region and authentication method (IAM and legacy authentication) which we have used for this prototype.
* Once the service is created click launch dashboard and create a Cloudant database(any suitable database name)
* Create service credentials for your Cloudant service by navigating service credentials=> new credentials=> Add.
* From the generated service credentials copy your Cloudant service URL(*https://\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*bluemix.cloudantnosqldb.appdomain.cloud*) which will be used later in IBM cloud function to connect to the Cloudant DB

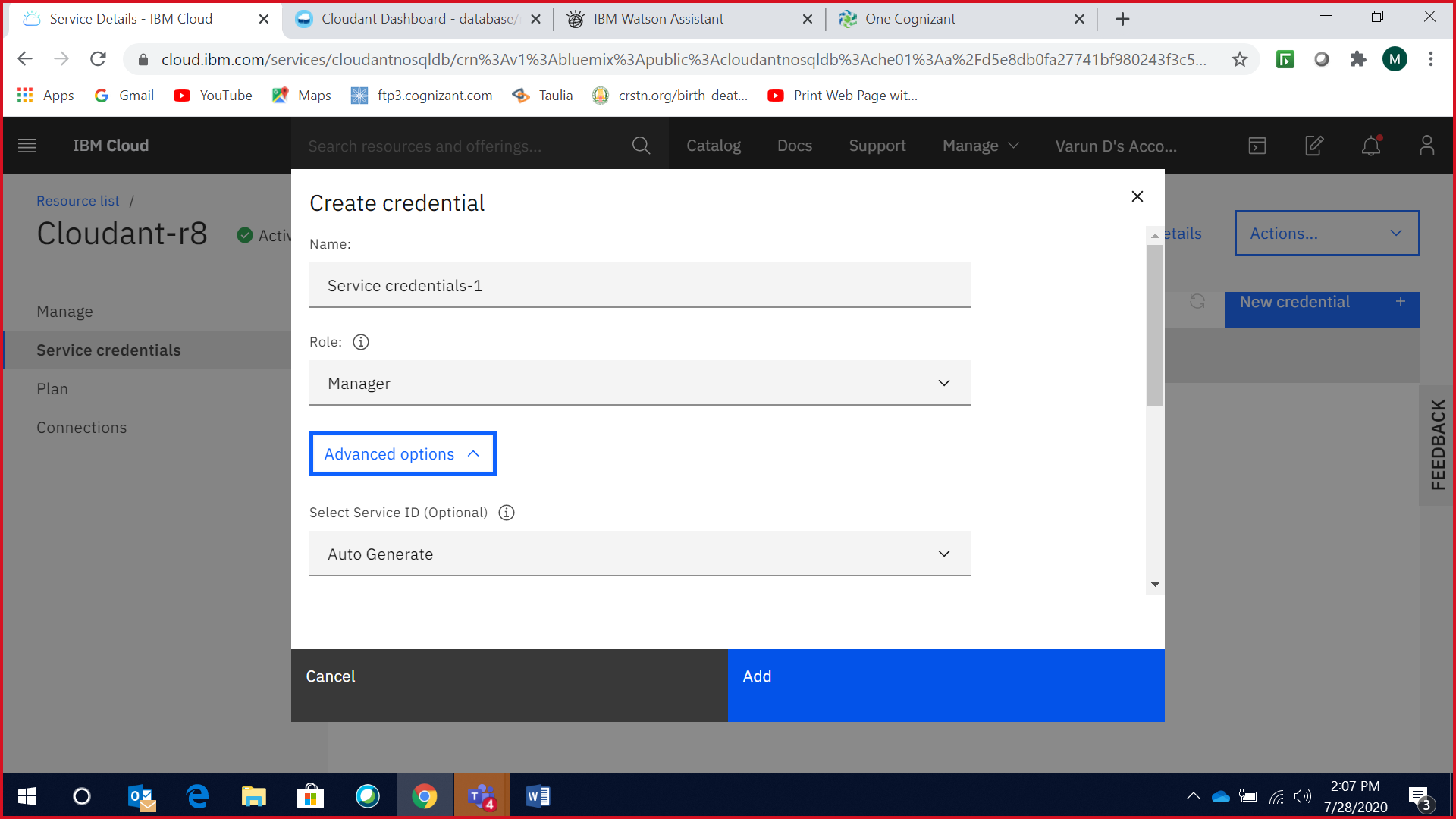
*Creating Cloudant service*



*Creating a database in IBM Cloudant*



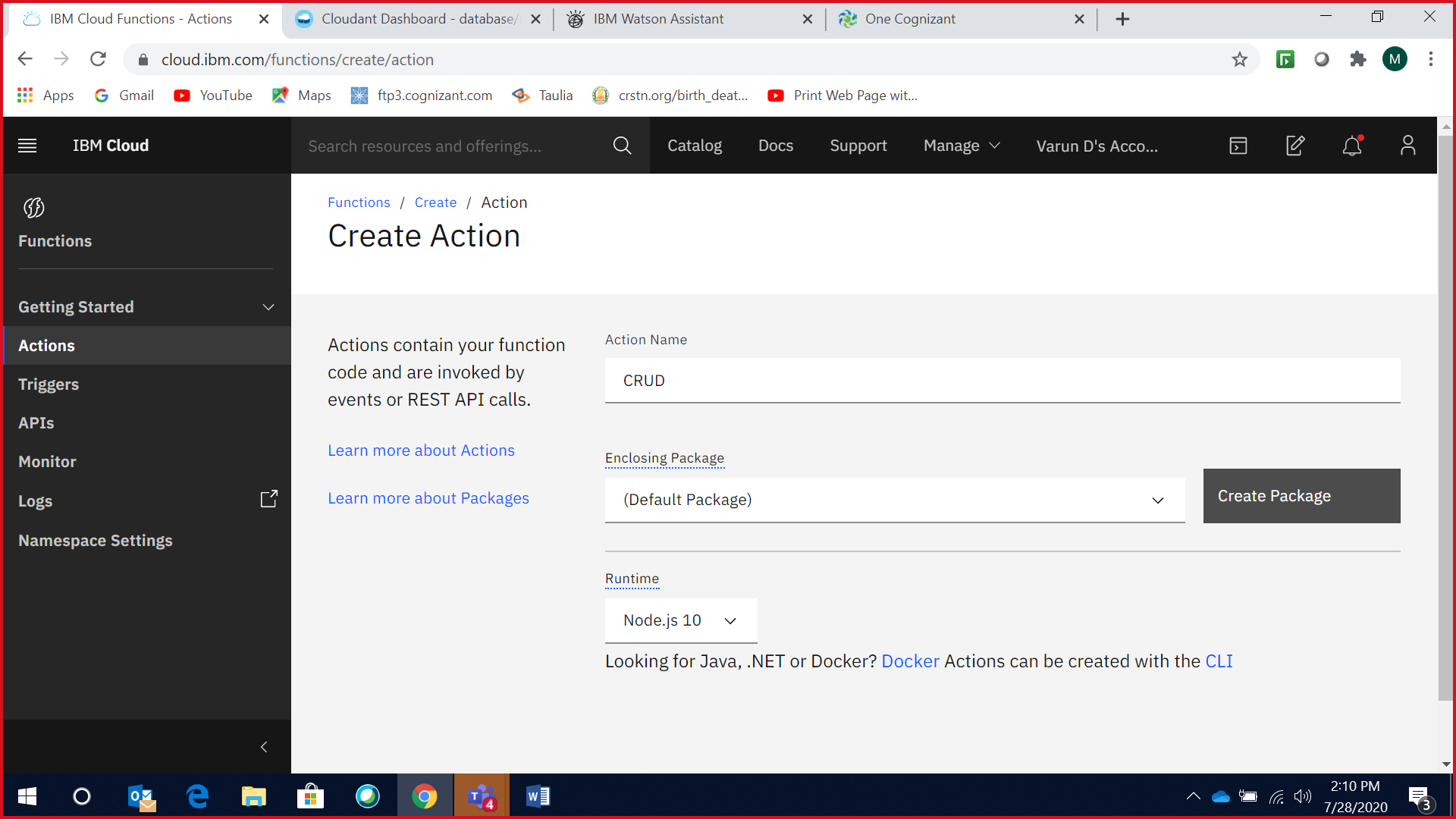
*Generating service credentials*



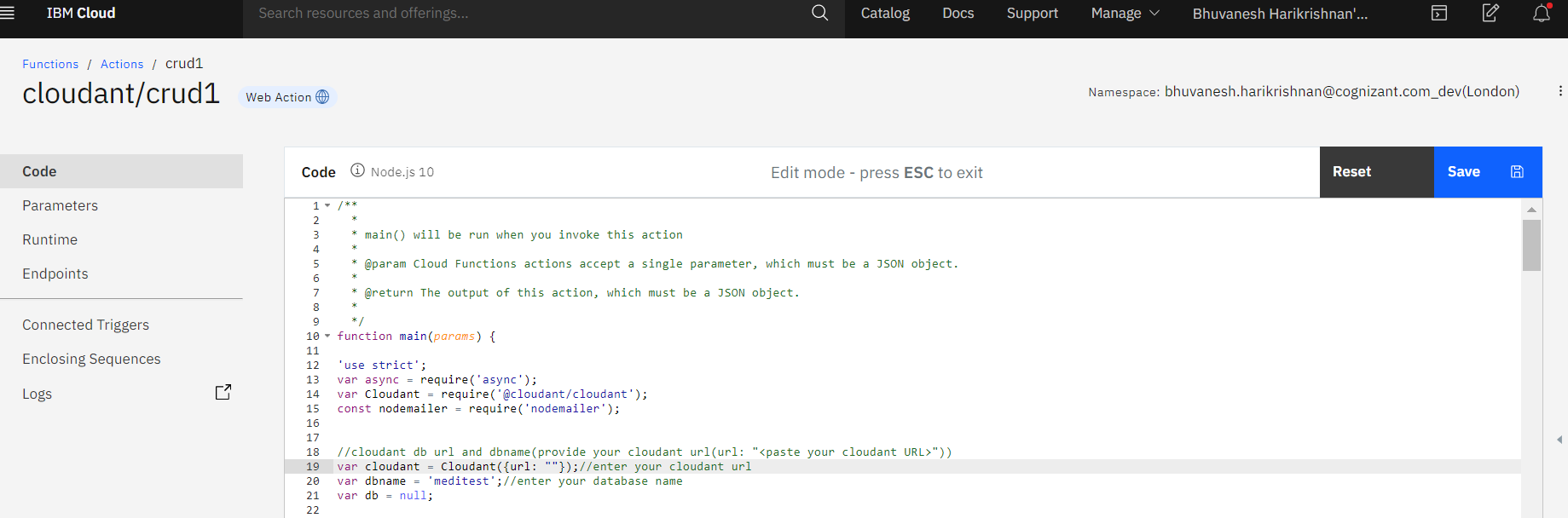
**Step 3:**

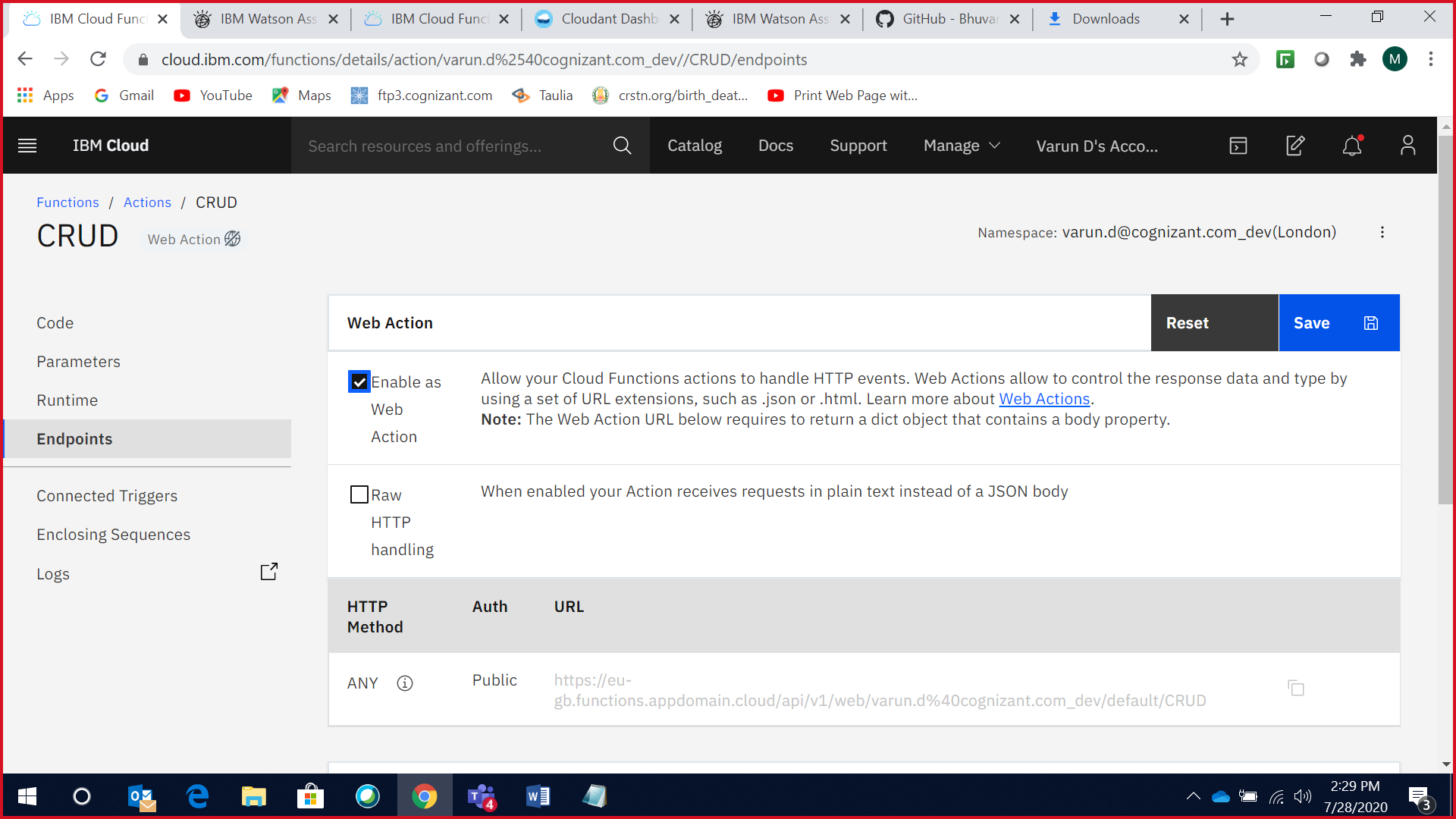
* Create an action in the IBM cloud function by providing an appropriate name and selecting the latest Node js version. In this prototype, we have selected Node.js 10.
* Copy the code from the Node.js file which we have extracted from Step 1 and paste the code to your IBM cloud function action which we have created in the above step.
* Now in the code enter the Cloudant URL which we would have retrieved from Step2 and enter the created database name to which we would connect and click Save.
* Navigate to Web Action in the created IBM cloud function action and click the checkbox ***'Enable as web action'***and copy the HTTP URL which will be used as Webhook in Watson Assistant to connect to the IBM Cloud Function action.

*Creating action in IBM Cloud Function*



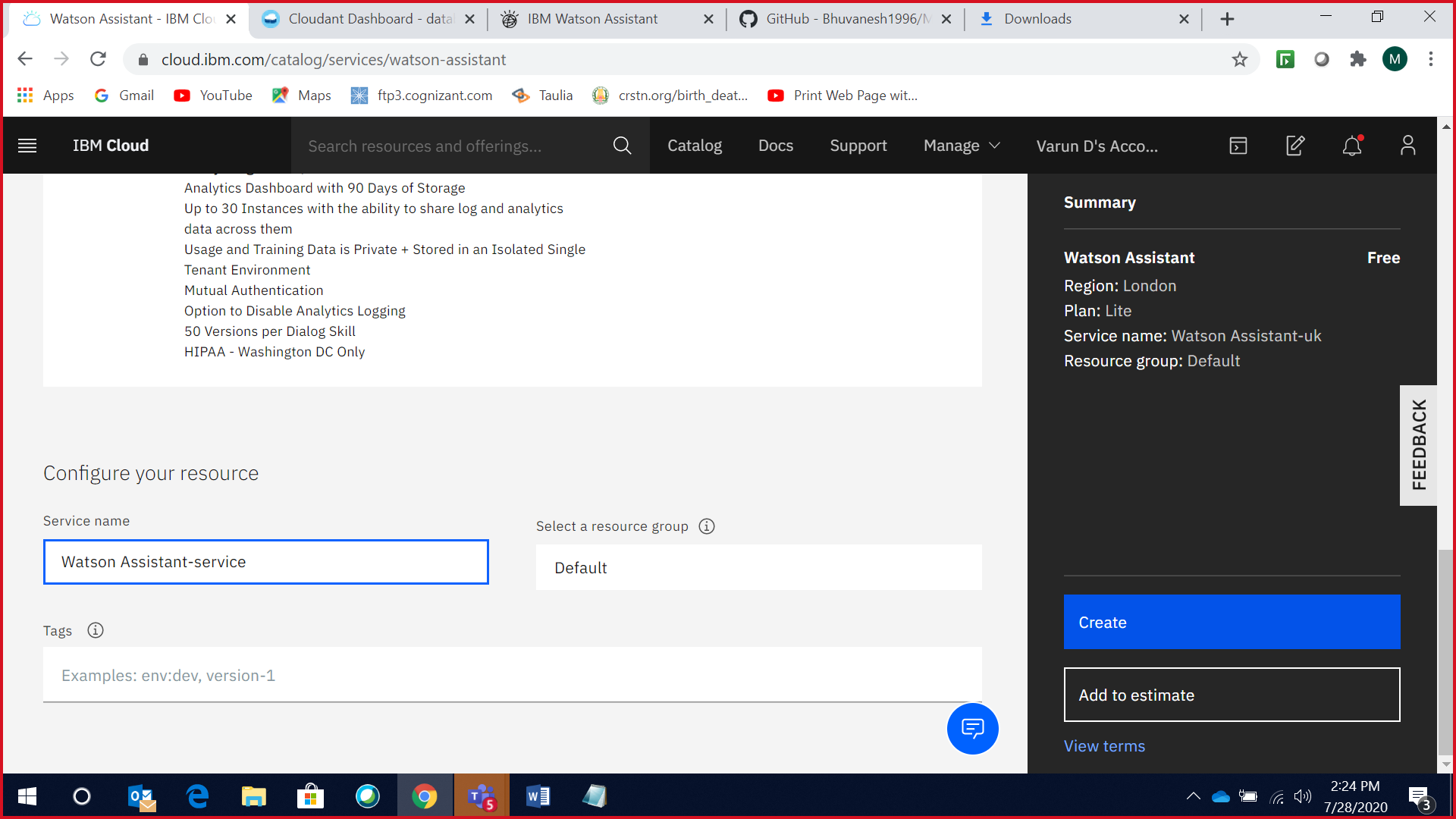
*Enter the cloudant URL in line no 19 and database name in line no 20 as shown below,*



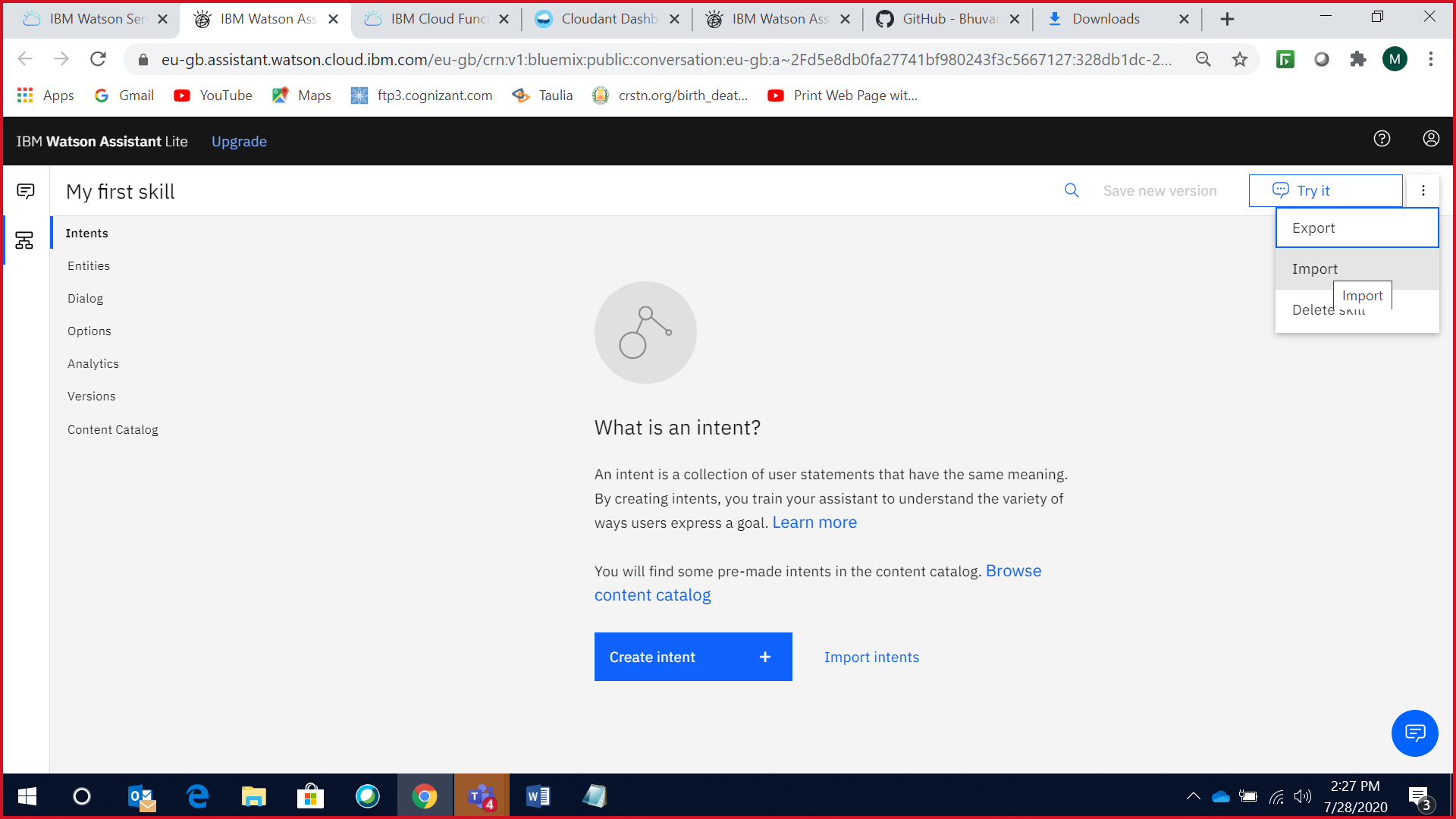
*Enable as WebAction and copy the HTTP Url as shown below,*

**Step 4:**

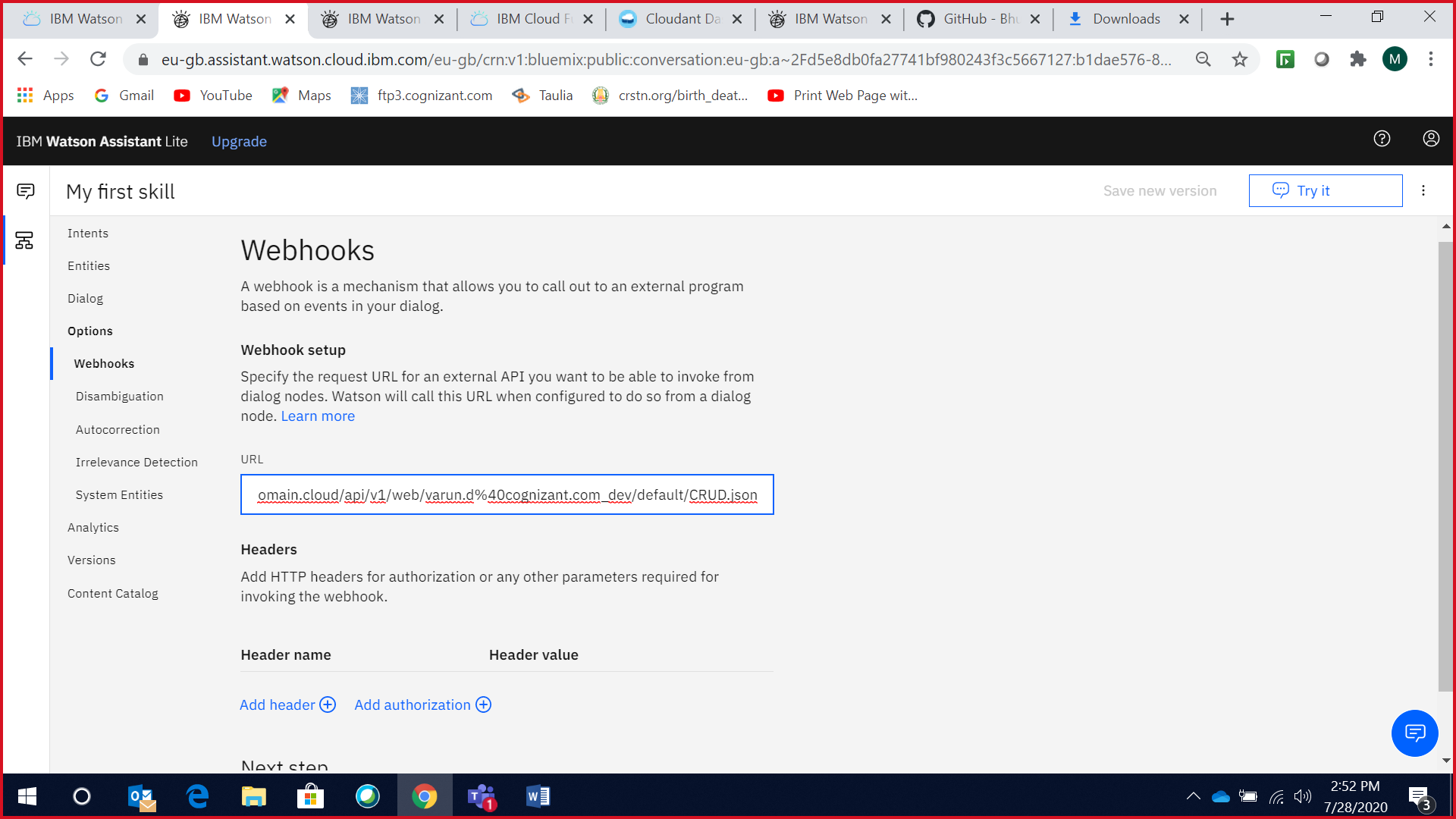
* Create Watson Assistant service, launch the assistant, and import the skill from the Skill-My-first-skill.json file which we have extracted in Step 1.
* Navigate to the options=>Webhooks in Watson Assistant and paste the Http URL which has been retrieved from Step 3 as our webhook and add .json extension to our webhook as the return type is JSON
* Now test the Watson Assistant in the try out pane.



*Importing Skill-My-first-skill file,*



*Add the URL from the above step as Webhook with the extension.json,*



**Step 5:**

**Email Notification:**

* Enter the From Mail id and its password in the Node.js code (IBM cloud function action) through which mail notification will be sent to the user once the appointment gets confirmed or canceled respectively.
* Make sure to Turn On the ‘Less secure app access’ for the Gmail account which will send mail notification by:

Login to your Gmail Account=>Manage your Google Acoount=>Security=>Turn On Less Secure App access.

* And also ‘Allow access to your Google account’ through the below link.

***https://accounts.google.com/DisplayUnlockCaptcha***

**Step 6:**

**Integration with Slack:**

* Refer to this URL for this slack integration(https://developer.ibm.com/tutorials/create-crisis-communication-chatbot-integrate-slack/)