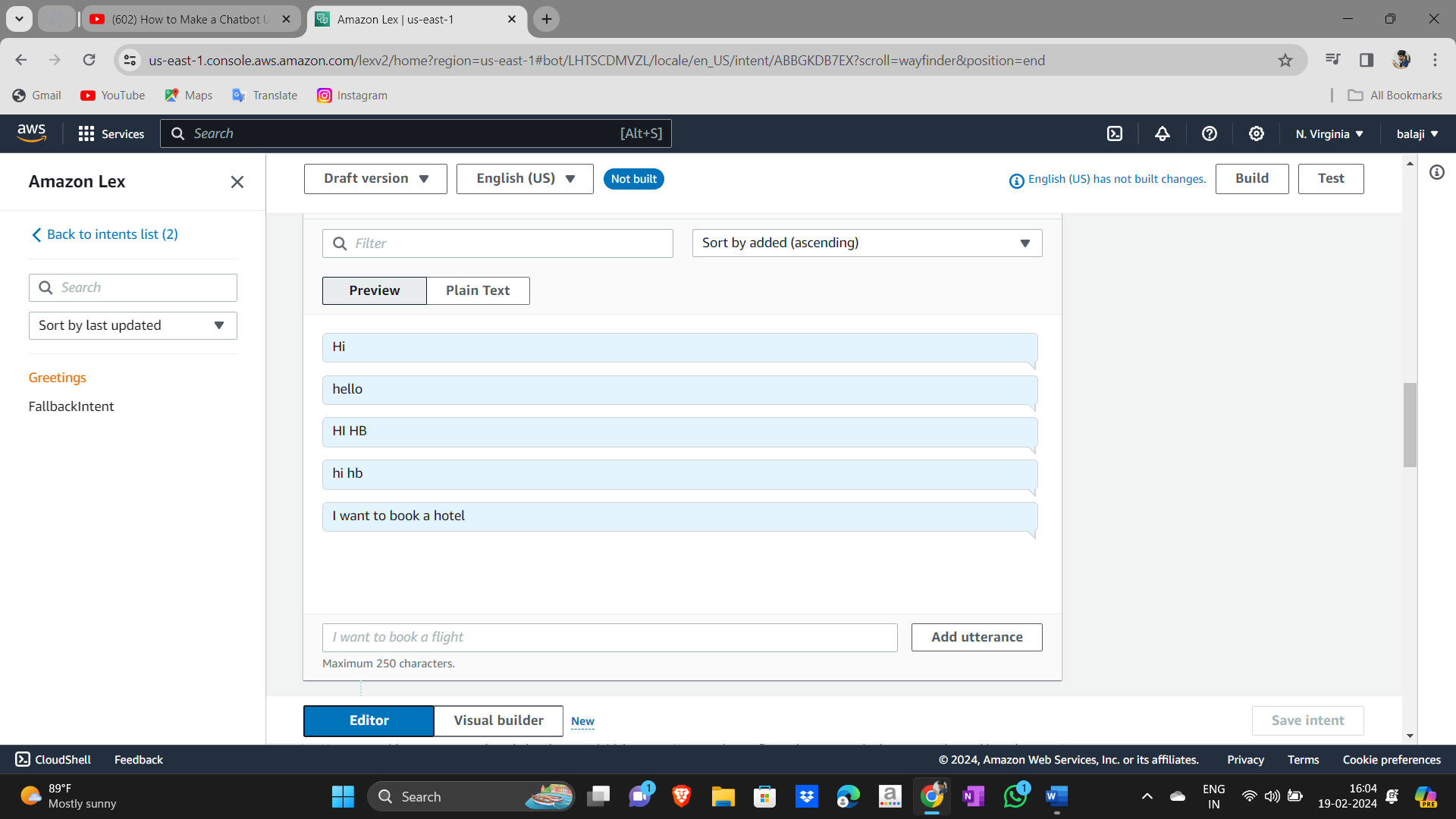
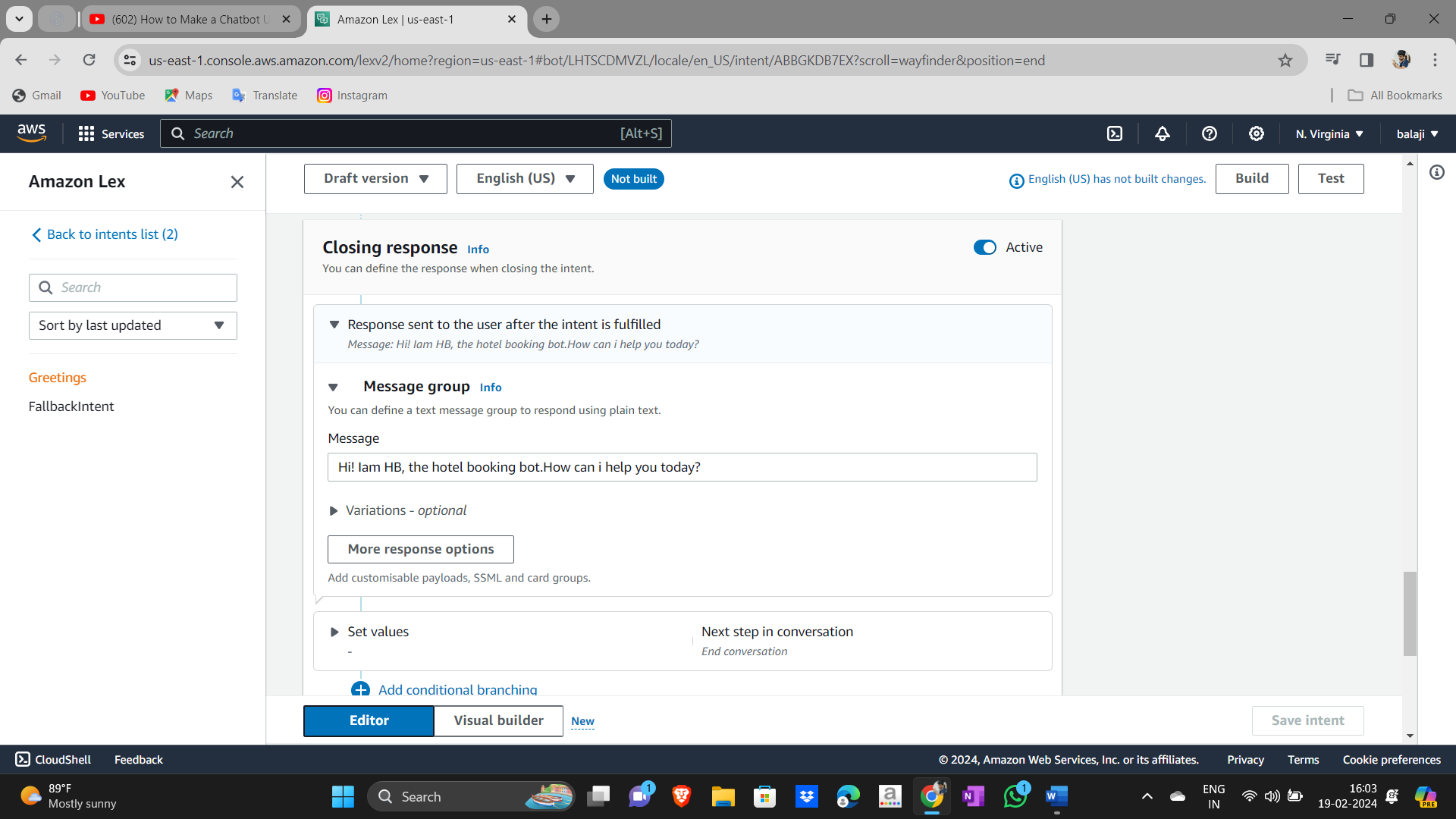
**AWS Lex Chatbot for Hotel Booking**

**Step 1: Create a Lex Bot**

1. Go to **AWS Console** and navigate to **Amazon Lex**.
2. Click on **Create** to create a new bot.
3. Choose the **Custom Bot** option and provide a name and description for your bot.
4. Configure the bot’s **language and voice**.
5. A screenshot of a computer

   AI-generated content may be incorrect.
6. Click **Done** to create the bot.

**Step 2: Create an Intent - Greetings**

1. Click on **Create Intent** and name it **Greetings**.
2. In the **Sample Utterances** section, add example phrases that users might say, such as:
   * "Hi"
   * "Hello"
   * "Good morning"
   * 
3. Scroll down to **Closing Response** and enter the bot’s response, such as:
   * "Hello! How can I assist you today?"
   * 
4. Click **Save Intent**.

**Step 3: Create an Intent - BookHotel**

1. Click **Create Intent** and name it **BookHotel**.
2. A screenshot of a computer

   AI-generated content may be incorrect.
3. Add **Sample Utterances**, such as:
   * "I want to book a hotel"
   * "Reserve a room for me"
   * A computer screen with a message box

     AI-generated content may be incorrect.
4. Click on **Slots** and add the following slots:
   * **NIGHTS** – Number of nights for the stay.
   * **CHECKIN\_DATE** – Check-in date.
   * **LOCATION** – City or location.
   * **ROOM\_TYPE** – Type of room (single, duplex, suite, etc.).
   * A screenshot of a computer

     AI-generated content may be incorrect.A screenshot of a computer

     AI-generated content may be incorrect.
5. To create **RoomType Slot Type**:
6. A screenshot of a computer

   AI-generated content may be incorrect.
   * Define values like "single", "duplex", "suite".
   * Save and use this slot type when adding **ROOM\_TYPE**.
   * A screenshot of a computer

     AI-generated content may be incorrect.
7. Click **Add Slot** and save the intent.
8. Click **Build the Bot**.

**Step 4: Configure Confirmation and Fulfillment**

1. Scroll down to **Confirmation Prompt** and define responses:
   * If the user confirms: "Great! Your room has been booked."
   * If the user declines: "Okay, let me know if you need assistance."
2. In **Fulfillment**, enter a prompt for successful and failed fulfillment.
3. In **Closing Response**, define a final response such as: "Thank you for using our service!"
4. Click **Save Intent** and **Build the Bot**.
5. A screenshot of a computer

   AI-generated content may be incorrect.

**Step 5: Create a Lambda Function**

1. Go to **AWS Lambda** in the AWS Console.
2. Click **Create Function** and name it **HotelBookingLambda**.
3. A screenshot of a computer

   AI-generated content may be incorrect.
4. Choose **Python** as the runtime.
5. Write the function logic (e.g., process booking requests, store details in a database, etc.).
6. Click **Deploy**.

**Step 6: Integrate Lambda with the Bot**

1. Under **Deployment**, click on **Aliases** → **Test bot-alias**.
2. Select the **Lambda Function** created earlier.
3. A computer screen with a message box

   AI-generated content may be incorrect.
4. Click **Save**.
5. Navigate to **BookHotel Intent** → **Code Hooks**.
6. Enable **Lambda Function** integration.
7. Click **Save Intent** and **Build the Bot**.
8. A screenshot of a computer

   AI-generated content may be incorrect.

**Step 7: Test the Bot**

1. Use the AWS Lex console to test different inputs.
2. Verify the responses for different scenarios:
   * Successful booking.
   * Missing slot values.
   * User declines confirmation.
3. Debug any issues and refine responses as needed.
4. A screenshot of a chat

   AI-generated content may be incorrect.A screenshot of a chat

   AI-generated content may be incorrect.
5. A screenshot of a chat

   AI-generated content may be incorrect.A screenshot of a chat

   AI-generated content may be incorrect.

**Conclusion**

You have successfully created a **hotel booking chatbot** using **AWS Lex and Lambda**! You can further enhance the bot by integrating it with:

* **Amazon Polly** for voice responses.
* **AWS DynamoDB** to store booking data.
* **Frontend UI (e.g., website or mobile app)** for better user experience.

**Next Steps:**

* Deploy the bot on a website.
* Add more intents, such as **Cancel Booking**.
* Improve conversation flow with better error handling.

🚀 **Congratulations! Your AWS Lex chatbot is now ready.**