|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Report:**Using Cloud Technology to Develop Chatbot | | | | | | | |  | |
|  | | | | | | | | | | | |
|  | Chatbot ID: @unlimitchatbot **<** [**https://github.com/Raymondwwf/COMP7940\_GroupProj\_40**](https://github.com/Raymondwwf/COMP7940_GroupProj_40) **>** | | | | | | | | | |  |
| Members of Group 40  Wong Tsz Kwan, 21456240 (Tempo)  Wong Wai Fung, 21437920 (Raymond)  Mo Lai Ting, 21464812 (Molly) | | | | | | Job Division  Tempo – App & Database Developer  Raymond – App & Database Developer  Molly – Preparation of report and presentation | | | | | |
|  | | | | | | | | | | | |
|  | | | | | | | | | | | |
| Why Implement Chatbot Anyway? | News & Insights | A&A Limited | Digital  Business ConsultingWhy Implement Chatbot Anyway? | News & Insights | A&A Limited | Digital  Business ConsultingFunctions  of our app  **Enquiry of Hiking Routes**  Users can select a district they are interested in, then our app will automatically and randomly select a hiking route from our database to users to start their hiking adventure.  **Photo & Video Sharing**  Users are free to share photos or video links of their hiking adventure through our app to other users. Users also do sharing and review via our app.  **“Hiking Tutor”**  Our app is pleased to hear users’ comments about our suggested hiking routes. Thus, users are allowed to comment and share their experience on the routes with other users, even act as a “hiking tutor”. Meanwhile, other users also are allowed to get “hiking tutors” telegram contact information through reviewing their sharing. | | |  |  | **Technical Specifications**   |  |  | | --- | --- | | **Components** | **Host** | | Database | MySQL 8.0 | | Auto Scaling | Amazon Elastic Compute Cloud (EC2) | | Load Balancer | AWS Gateway Load Balancer | | CI/CD | GitHub Action | | Container Services | Docker |   **Summary of our app**   |  |  | | --- | --- | | **Command** | **Description** | | /start | Display our feature menu to user | | /cookshare | Allow users to share their cooking video | | /sharemovie | Allow users to share their movie review of specific movie | | /seemoviecomment | Display users’ review of specific movie | | /hikeshare | Allow to input users’ experience and photos for our suggested hiking route, and record their user ID | | /skipshare | Skip sharing photo process | | /viewhikeshare | Recall users’ sharing and provide sharers’ contact to build connection | | | | | | | |
|
| Database System  Our app is designed to have interactions between different users which allows them to share and view, even interact between users.  Thus, our app required MySQL database system which can set relationships between databases to recall shared information or users’ telegram contact. Also, our app allows user to share different information, including text, image or video link so a great storage capability should be needed. | |  |  | **Cloud Platform**  Our app uses Amazon Web Services (AWS) as our cloud service provider since it provides various services to support our app.  EC2 of AWS provides us a virtual server to run, deploy, monitor and maintain our app. AMI of EC2 also has a strong scalability allows various deployments of application, including Docker, MySQL, GitHub, etc. | | |

|  |  |  |
| --- | --- | --- |
|  | @unlimitchatbot |  |
|  | | |
| **Data Flow**  If a user sends a message to our chatbot, then Telegram will automatically generate a request to our back-end (EC2) for getting data from our MySQL database by our Python app, MySQL will return data to the app and generate a response to Telegram, and display it to the user. | | |
| **Workflow**    If there is any push action to GitHub from our App developer, then it will trigger GitHub Action to use SSH to deploy our updated code to EC2 automatically and run deploy script for EC2. | | |
| **Multiple Container**    Our app uses multiple containers to build our MySQL database and Python app. It can provide two isolated environments for our developer to do configuration much easier and quickly. Meanwhile, EC2 also allow us to build our app by working with multiple containers. | | |
|  | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **@unlimitchatbot** | | | | | |
|  | | | | | | |
|  | | | | | | |
| **Load Balancer**    To prevent high traffic load to breakdown our app, AWS Gateway Load Balancer is used for controlling our traffic load balance. All our chatbot users are specified by instance ID and their requests are required to enter our app before our port. GWLBE will monitor the traffic of target group, maintain the traffic at acceptable level and allow them to enter our AWS severs. It helps our app to enhance the security level and compliance to avoid hackers’ attack and maintain the stability of our app. | | | | | |
| **Evidence of Fulfillment** | | | | | | |
|  | | | | | | |
|  | |  |  | References   1. https://farhan-tanvir.medium.com/ci-cd-from-github-to-aws-ec2-using-github-action-e18b621c0507 2. https://aws.amazon.com/blogs/containers/create-a-ci-cd-pipeline-for-amazon-ecs-with-github-actions-and-aws-codebuild-tests/ 3. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/create-key-pairs.html |