Docker Deployment Document

1. Objective

This document describes the Deployment Details of AI-powered Rasa Chatbot. It focuses on building the necessary Docker images, composing the network, and creating containers. This document is meant for the Software Developer and Maintenance Team members.

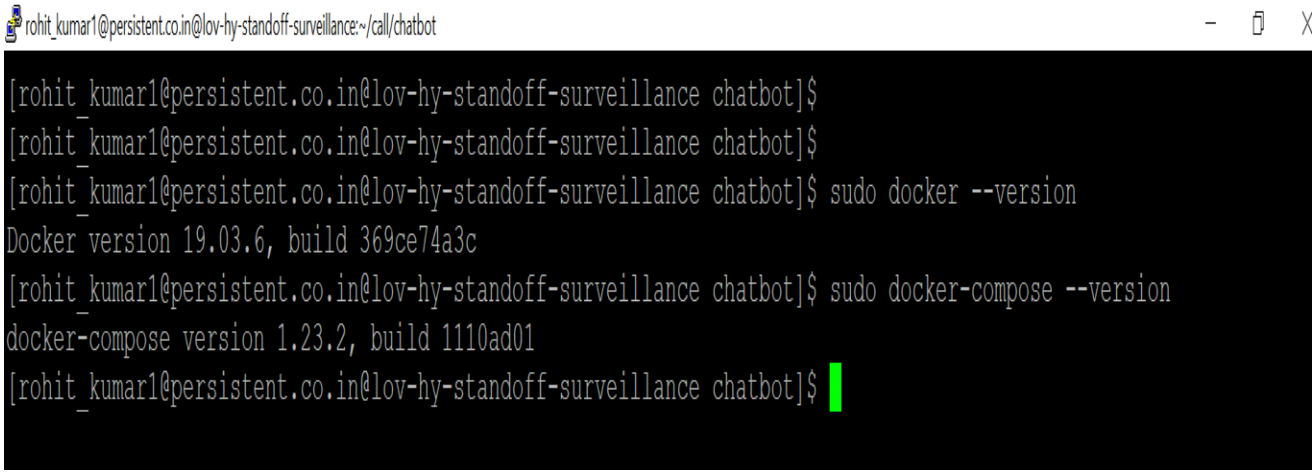
1.1 About Rasa Chatbot

The AI-Powered Rasa Chatbot makes use of two services/servers, specifically Rasa Server and the Action Server. Docker Compose allows us to run both of the servers simultaneously while interacting with each other, maintain the chat history in the container, expose the host over API, and many other operations.

2. Prerequisites

Docker (version 18 or above) and Docker Compose must be installed on the Server/VM.

- To check Docker version: **docker --version**
- To check docker-compose version: **docker-compose --version**



3. How to get started

This section covers the steps needed to create the Compose network, set up both servers, and set up the intercommunication between them.

3.1 Clone the Repository

Clone the repository from the GitHub: **git clone** https://github.com/NSSAC/chatbot.git .In order to clone you must have the access to the NSSAC/chatbot repository.

```
[rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance awe]$ git clone https://github.com/NSSAC/chatbot.git
Cloning into 'chatbot'...
Username for 'https://github.com': rohit-persistent
Password for 'https://rohit-persistent@github.com':
remote: Enumerating objects: 2095, done.
remote: Counting objects: 100% (2095/2095), done.
remote: Compressing objects: 100% (1763/1763), done.
remote: Total 2095 (delta 337), reused 2072 (delta 324), pack-reused 0
Receiving objects: 100% (2095/2095), 43.68 MiB | 9.59 MiB/s, done.
Resolving deltas: 100% (337/337), done.
[rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance awe]$ cd chatbot/
[rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance chatbot]$
```

The cloned repository will have Dockerfile and docker-compose.yml file which contains all the rasa image, rasa-dependencies, network and volumes information. Below is the snapshot of the Dockerfile:

```
rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance:~/git_rasa

#Base image
FROM ubuntu

#Extend the official Rasa SDK image
FROM rasa/rasa-sdk:1.10.2

MAINTAINER "Rohit Kumar"

#Change back to root user to install dependencies
USER root

#Work Directory
WORKDIR /app

#Installing git and cloning the data from the gitHub
RUN apt-get update && apt-get upgrade -y &&  apt-get install -y git
RUN git clone  https://2b863d8ba6c117a331f107e08241d33a95c572a2:x-oauth-basic@github.com/NSSAC/chatbot.git

#Adding the codebase to our work directory
RUN cp -R ./chatbot/* ./

# Install extra requirements for actions code, if necessary (uncomment next line)
RUN pip install -U pip && pip install word2number && pip install dateparser
RUN pip3 install pandas

# By best practices, don't run the code with root user
USER 1001

CMD ["start", "--actions", "actions", "--debug"]
~
"Dockerfile" 29L, 801C                                                    1,1           All
```

docker-compose.yml file is responsible for the internal port mappings, creating a compose network for both action and rasa server, and the volumes information.

Below is the snapshot of the docker-compose.yml file:

rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance:~/awe/chatbot

```
version: '3.0'
services:
  rasa:
    image: rasa/rasa:1.10.11-full
    ports:
      - 5005:5005
    volumes:
      - ./:/app
    command:
      - run
      - --enable-api
      - --endpoints
      - endpoints.yml
      - --cors
      - "*"

  app:
    image: rasa/rasa-action-server
    volumes:
      - ./actions:/app/actions
    ports:
      - 5055:5055

~
~
~
"docker-compose.yml" 23L, 358C
```

3.2 Actions after cloning the repository

- Go to the cloned directory 'chatbot' where Dockerfile and docker-compose.yml file exists
- Run **docker build -t rasa/rasa-action-server .** at the level where the Dockerfile exists. It will create a custom rasa action server image. The name of the image must be **rasa/rasa-action-server** or, if you want a custom image name, you also need to change the image name in the docker-compose.yml file. You can see the images by running the **docker images** command.

- As in the snapshot below, rasa/rasa-action-server is our image.

```
[rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance chatbot]$ sudo docker images
REPOSITORY                TAG                 IMAGE ID            CREATED              SIZE
rasa/rasa-action-server   latest              ec7eb136852a        About a minute ago   707MB
python                    latest              28a4c88cdbbf        2 weeks ago          882MB
rasa/rasa                 1.10.11-full        e3da25b13223        4 weeks ago          2.72GB
ubuntu                    latest              4e2eef94cd6b        5 weeks ago          73.9MB
rasa/rasa-sdk             1.10.2              97eb780f0ce2        3 months ago         196MB
```

- Now, run **docker-compose up** to start both rasa and action servers. This command will create an internal network, start both the servers simultaneously and interconnect them and you will be able to see the logs on the console. In order to run rasa in the background refer section 6.

```
[rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance chatbot]$ sudo docker-compose up
Creating network "chatbot_default" with the default driver
Creating chatbot_rasa_1 ... done
Creating chatbot_app_1  ... done
Attaching to chatbot_app_1, chatbot_rasa_1
app_1   | 2020-09-25 10:49:21 INFO     rasa_sdk.endpoint  - Starting action endpoint server...
app_1   | 2020-09-25 10:49:21 INFO     rasa_sdk.executor  - Registered function for 'action_count'.
app_1   | 2020-09-25 10:49:21 INFO     rasa_sdk.endpoint  - Action endpoint is up and running on http://localhost:5055
app_1   | 2020-09-25 10:49:21 DEBUG    rasa_sdk.utils  - Using the default number of Sanic workers (1).
rasa_1  | 2020-09-25 10:49:38 INFO     root  - Starting Rasa server on http://localhost:5005
rasa_1  | 2020-09-25 10:50:08 INFO     rasa.nlu.components  - Added 'SpacyNLP' to component cache. Key 'SpacyNLP-en_core_web_md'.
rasa_1  | 2020-09-25 10:50:08.958140: E tensorflow/stream_executor/cuda/cuda_driver.cc:351] failed call to cuInit: UNKNOWN ERROR (303)
rasa_1  | /opt/venv/lib/python3.7/site-packages/sklearn/externals/joblib/__init__.py:15: FutureWarning: sklearn.externals.joblib is deprecated in 0.21 and wi
ll be removed in 0.23. Please import this functionality directly from joblib, which can be installed with: pip install joblib. If this warning is raised when
 loading pickled models, you may need to re-serialize those models with scikit-learn 0.21+.
rasa_1  |   warnings.warn(msg, category=FutureWarning)
rasa_1  | 2020-09-25 10:50:23 INFO     root  - Rasa server is up and running.
```

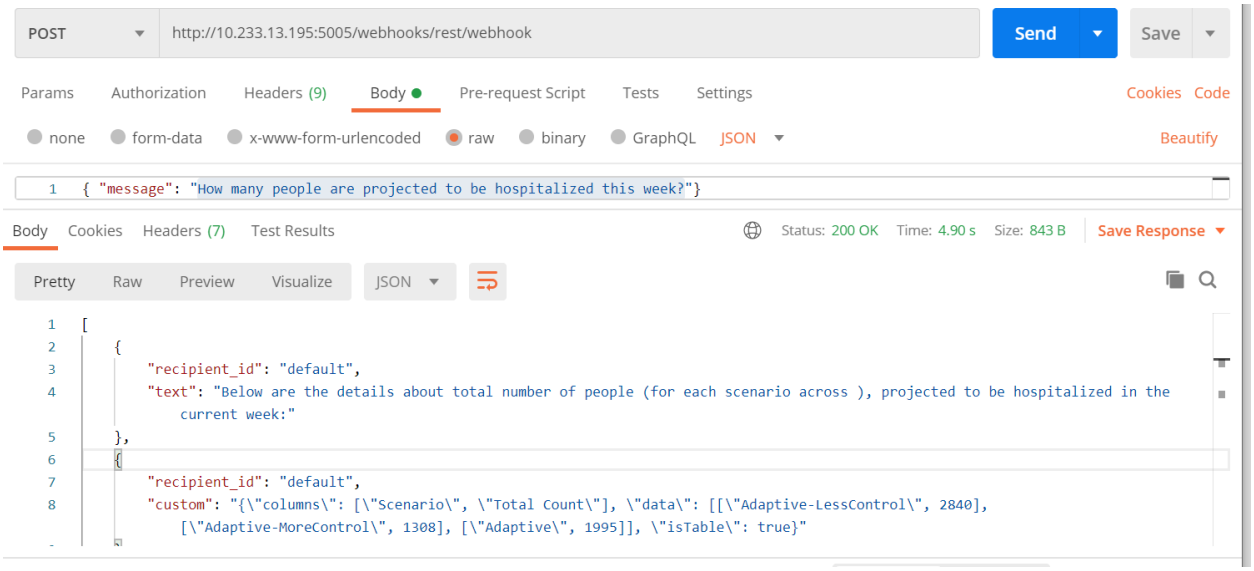
4. Verification/Validation

4.1 Validate using Post API

Now that both of the servers are up and running, we can access the docker service by making a POST API request to the docker container using the details below:

- http://<host-ip>:5005/webhooks/rest/webhook
- Content-type: application/json
- Body example:

  {“message”: “How many people are projected to be hospitalized this week?”}



4.2 Validate using curl on console

We can also verify by sending the messages to the bot from the console as shown below:

```
curl --request POST \

  --url http://localhost:5005/webhooks/rest/webhook \

  --header 'content-type: application/json' \

  --data '{

    "message": "How many people are projected to be hospitalized this week?"

  }'
```

Your chatbot should answer similar to this:


```
[{"recipient_id":"default","text":"Below are the details about total number of people
(for each scenario across ), projected to be hospitalized in the current week:"},

{"recipient_id":"default","custom":"{\"columns\": [\"Scenario\", \"Total Count\"],
\"data\": [[\"Adaptive-LessControl\", 2840], [\"Adaptive-MoreControl\", 1308],
[\"Adaptive\", 1995]], \"isTable\": true}"},

{"recipient_id":"default","text":"What else do you want to check about?",

"buttons":[{"payload":"\/hospitalization","title":"Hospitalization"},

{"payload":"\/occupied_beds","title":"Occupied
Beds"},{"payload":"\/about_mrdd","title":"About MRDD"}]
```


4.2 Verification from running containers

To list out all of the running containers, run the **docker ps –a** command. In the snapshot below, both the rasa-action-server and rasa server containers can be seen up and running

```
[rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance files]$
[rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance files]$
[rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance files]$
[rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance files]$ sudo docker ps -a
CONTAINER ID        IMAGE                   COMMAND                  CREATED             STATUS              PORTS                    NAMES
c5a3440d93ea        rasa/rasa-action-server  "./entrypoint.sh sta…"   About a minute ago  Up About a minute   0.0.0.0:5055->5055/tcp   chatbot_app_1
676a163140ee        rasa/rasa:1.10.11-full   "rasa run --enable-a…"   About a minute ago  Up About a minute   0.0.0.0:5005->5005/tcp   chatbot_rasa_1
[rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance files]$
```

5. Container actions

5.1 Inside container

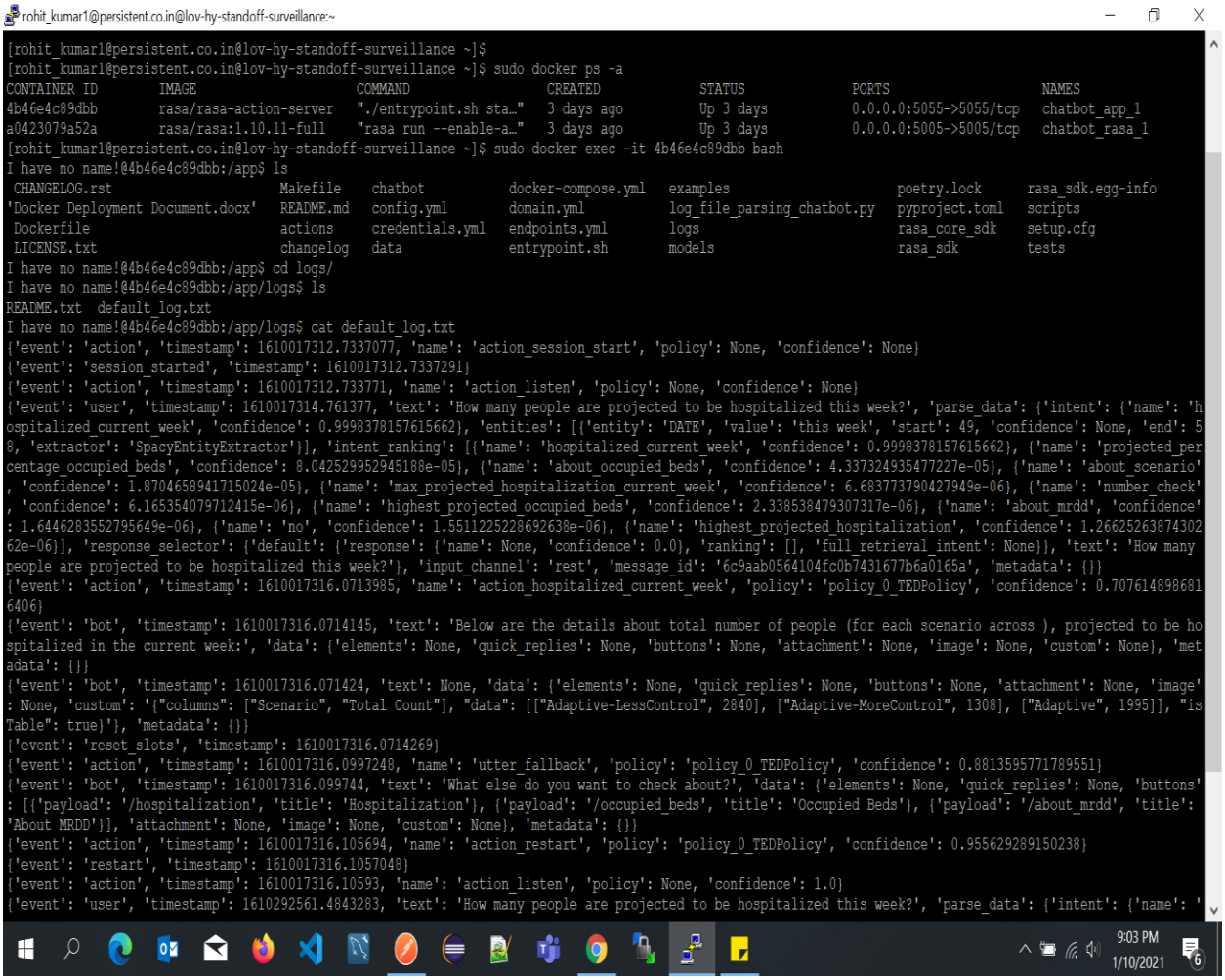
To work inside the running container, run **docker exec -it <rasa-action-server-container-id> bash**

```
[rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance files]$ sudo docker exec -it c5a3440d93ea bash
I have no name!@c5a3440d93ea:/app$ ls
CHANGELOG.rst            Makefile     changelog        data               entrypoint.sh  poetry.lock     rasa_sdk.egg-info
CustomEntityExtractor.py README.md    chatbot          docker-compose.yml examples       pyproject.toml  scripts
Dockerfile               __init__.py  config.yml       domain.yml         listfile.txt   rasa_core_sdk   setup.cfg
LICENSE.txt              actions      credentials.yml  endpoints.yml      models         rasa_sdk        tests
I have no name!@c5a3440d93ea:/app$
```

5.2 chat history inside container

Chat history with chatbot is maintained in the directory named logs inside in the rasa action container, and to see the contents of the chat history we need to go to that directory and see as shown in the snapshot below.

```
[rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance ~]$
[rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance ~]$ sudo docker ps -a
CONTAINER ID        IMAGE                    COMMAND                  CREATED             STATUS              PORTS                    NAMES
4b46e4c89dbb        rasa/rasa-action-server  "./entrypoint.sh sta…"   3 days ago          Up 3 days           0.0.0.0:5055->5055/tcp   chatbot_app_1
a0423079a52a        rasa/rasa:1.10.11-full   "rasa run --enable-a…"   3 days ago          Up 3 days           0.0.0.0:5005->5005/tcp   chatbot_rasa_1
[rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance ~]$ sudo docker exec -it 4b46e4c89dbb bash
I have no name!@4b46e4c89dbb:/app$ ls
 CHANGELOG.rst                   Makefile    chatbot          docker-compose.yml  examples                    poetry.lock     rasa_sdk.egg-info
'Docker Deployment Document.docx' README.md  config.yml       domain.yml          log_file_parsing_chatbot.py pyproject.toml  scripts
 Dockerfile                      actions     credentials.yml  endpoints.yml       logs                        rasa_core_sdk   setup.cfg
 LICENSE.txt                     changelog   data             entrypoint.sh       models                      rasa_sdk        tests
I have no name!@4b46e4c89dbb:/app$ cd logs/
I have no name!@4b46e4c89dbb:/app/logs$ ls
README.txt  default_log.txt
I have no name!@4b46e4c89dbb:/app/logs$ cat default_log.txt
```



```
rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance:~
[rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance ~]$
[rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance ~]$ sudo docker ps -a
CONTAINER ID        IMAGE                    COMMAND                  CREATED             STATUS              PORTS                    NAMES
4b46e4c89dbb        rasa/rasa-action-server  "./entrypoint.sh sta…"   3 days ago          Up 3 days           0.0.0.0:5055->5055/tcp   chatbot_app_1
a0423079a52a        rasa/rasa:1.10.11-full   "rasa run --enable-a…"   3 days ago          Up 3 days           0.0.0.0:5005->5005/tcp   chatbot_rasa_1
[rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance ~]$ sudo docker exec -it 4b46e4c89dbb bash
I have no name!@4b46e4c89dbb:/app$ ls
 CHANGELOG.rst                    Makefile    chatbot          docker-compose.yml   examples                     poetry.lock      rasa_sdk.egg-info
'Docker Deployment Document.docx'  README.md   config.yml       domain.yml           log_file_parsing_chatbot.py  pyproject.toml   scripts
 Dockerfile                       actions     credentials.yml  endpoints.yml        logs                         rasa_core_sdk    setup.cfg
 LICENSE.txt                      changelog   data             entrypoint.sh        models                       rasa_sdk         tests
I have no name!@4b46e4c89dbb:/app$ cd logs/
I have no name!@4b46e4c89dbb:/app/logs$ ls
README.txt  default_log.txt
I have no name!@4b46e4c89dbb:/app/logs$ cat default_log.txt
{'event': 'action', 'timestamp': 1610017312.7337077, 'name': 'action_session_start', 'policy': None, 'confidence': None}
{'event': 'session_started', 'timestamp': 1610017312.7337291}
{'event': 'action', 'timestamp': 1610017312.733771, 'name': 'action_listen', 'policy': None, 'confidence': None}
{'event': 'user', 'timestamp': 1610017314.761377, 'text': 'How many people are projected to be hospitalized this week?', 'parse_data': {'intent': {'name': 'h
ospitalized_current_week', 'confidence': 0.9998378157615662}, 'entities': [{'entity': 'DATE', 'value': 'this week', 'start': 49, 'confidence': None, 'end': 5
8, 'extractor': 'SpacyEntityExtractor'}], 'intent_ranking': [{'name': 'hospitalized_current_week', 'confidence': 0.9998378157615662}, {'name': 'projected_per
centage_occupied_beds', 'confidence': 8.042529952945188e-05}, {'name': 'about_occupied_beds', 'confidence': 4.337324935477227e-05}, {'name': 'about_scenario'
, 'confidence': 1.8704658941715024e-05}, {'name': 'max_projected_hospitalization_current_week', 'confidence': 6.683773790427949e-06}, {'name': 'number_check'
, 'confidence': 6.165354079712415e-06}, {'name': 'highest_projected_occupied_beds', 'confidence': 2.338538479307317e-06}, {'name': 'about_mrdd', 'confidence'
: 1.6446283552795649e-06}, {'name': 'no', 'confidence': 1.5511225228692638e-06}, {'name': 'highest_projected_hospitalization', 'confidence': 1.26625263874302
62e-06}], 'response_selector': {'default': {'response': {'name': None, 'confidence': 0.0}, 'ranking': [], 'full_retrieval_intent': None}}, 'text': 'How many
people are projected to be hospitalized this week?'}, 'input_channel': 'rest', 'message_id': '6c9aab0564104fc0b7431677b6a0165a', 'metadata': {}}
{'event': 'action', 'timestamp': 1610017316.0713985, 'name': 'action_hospitalized_current_week', 'policy': 'policy_0_TEDPolicy', 'confidence': 0.707614898681
6406}
{'event': 'bot', 'timestamp': 1610017316.0714145, 'text': 'Below are the details about total number of people (for each scenario across ), projected to be ho
spitalized in the current week:', 'data': {'elements': None, 'quick_replies': None, 'buttons': None, 'attachment': None, 'image': None, 'custom': None}, 'met
adata': {}}
{'event': 'bot', 'timestamp': 1610017316.071424, 'text': None, 'data': {'elements': None, 'quick_replies': None, 'buttons': None, 'attachment': None, 'image'
: None, 'custom': '{"columns": ["Scenario", "Total Count"], "data": [["Adaptive-LessControl", 2840], ["Adaptive-MoreControl", 1308], ["Adaptive", 1995]], "is
Table": true}'}, 'metadata': {}}
{'event': 'reset_slots', 'timestamp': 1610017316.0714269}
{'event': 'action', 'timestamp': 1610017316.0997248, 'name': 'utter_fallback', 'policy': 'policy_0_TEDPolicy', 'confidence': 0.8813595771789551}
{'event': 'bot', 'timestamp': 1610017316.099744, 'text': 'What else do you want to check about?', 'data': {'elements': None, 'quick_replies': None, 'buttons'
: [{'payload': '/hospitalization', 'title': 'Hospitalization'}, {'payload': '/occupied_beds', 'title': 'Occupied Beds'}, {'payload': '/about_mrdd', 'title':
'About MRDD'}], 'attachment': None, 'image': None, 'custom': None}, 'metadata': {}}
{'event': 'action', 'timestamp': 1610017316.105694, 'name': 'action_restart', 'policy': 'policy_0_TEDPolicy', 'confidence': 0.955629289150238}
{'event': 'restart', 'timestamp': 1610017316.1057048}
{'event': 'action', 'timestamp': 1610017316.10593, 'name': 'action_listen', 'policy': None, 'confidence': 1.0}
{'event': 'user', 'timestamp': 1610292561.4843283, 'text': 'How many people are projected to be hospitalized this week?', 'parse_data': {'intent': {'name': '
```

5.3 stopping the containers

- To stop the compose network/servers, simply run **ctrl+c**; to remove all volume information, run **docker-compose down** command.
- And to stop the containers, run: **docker stop <container-id>**

6. Running docker process in background

To run the docker compose network in the background run **docker-compose up -d**

```
[rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance thursdaycall]$
[rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance thursdaycall]$ sudo docker-compose down
Stopping thursdaycall_rasa_1 ... done
Stopping thursdaycall_app_1  ... done
Removing thursdaycall_rasa_1 ... done
Removing thursdaycall_app_1  ... done
Removing network thursdaycall_default
[rohit_kumar1@persistent.co.in@lov-hy-standoff-surveillance thursdaycall]$ sudo docker ps -a
```

- To stop this demon process or running rasa in the background run **docker-compose down**
- As it is running in the background, logs of the rasa servers will not appear on the console. If you still want to see the logs of the containers run **docker container logs <container-id>**

---

7. Rasa Training

To train the Rasa for the newly added intents, or if you don't have any model available to you, then run

docker run --user 1000 -it -v $(pwd):/app rasa/rasa:1.10.11-full train

---

8. Re-Deploying in the case of new commits

- If there is a new commit GitHub, then follow the instructions below:
  - Stop the currently running compose network: **ctrl+c**
  - Delete the old rasa-action-server image: **docker rmi rasa/rasa-action-server**
  - Run **docker build -t rasa/rasa-action-server .**
  - Then run **docker-compose up**

- **docker system prune** command removes all the dangling images and stopped containers.
- To inspect or learn more information about a container like ipaddress,port,volume etc.

  run docker inspect <container-id>

- Point to remember: Always run build and compose commands at the level where the Dockerfile and docker-compose.yml file is present.
