

Docker Deployment Document

****

[1. Objective 3](#_Toc53484461)

[1.1 About Rasa Chatbot 3](#_Toc53484462)

[2. Prerequisites 3](#_Toc53484463)

[3. How to get Started 4](#_Toc53484464)

[3.1 Clone the Repository 4](#_Toc53484465)

[3.2 Actions after cloning the repository 6](#_Toc53484466)

[4. Verification/Validation 7](#_Toc53484467)

[4.1 Validate using Post API 7](#_Toc53484468)

[4.2 Verification from running containers 7](#_Toc53484469)

[5. Container actions 8](#_Toc53484470)

[5.1 Inside container 8](#_Toc53484471)

[5.2 chat history inside container 8](#_Toc53484472)

[5.3 stopping the containers 9](#_Toc53484473)

[6. Rasa Train 9](#_Toc53484474)

[7. Re-Deploying in case of new commits 9](#_Toc53484475)

# Objective

This document describes the Deployment Details of AI powered rasa chatbot. It focuses on Building the images, compose network and creating containers .This document is meant for the Software Developer and Maintenance Team.

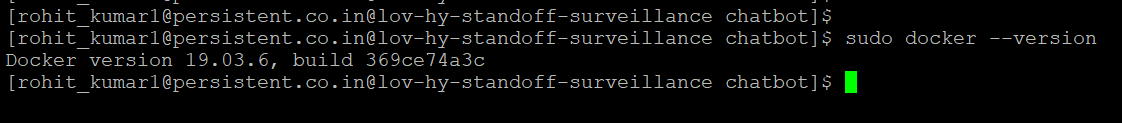
## About Rasa Chatbot

The AI-Powered rasa chatbot makes uses of two services/servers i.e., Rasa Server and the Action Server .Docker compose allows us to run both the servers simultaneously and interact with each other ,maintain the chat history in the container , expose the host over API and many other operations.

# Prerequisites

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

By running docker –version we can verify docker and docker compose version.

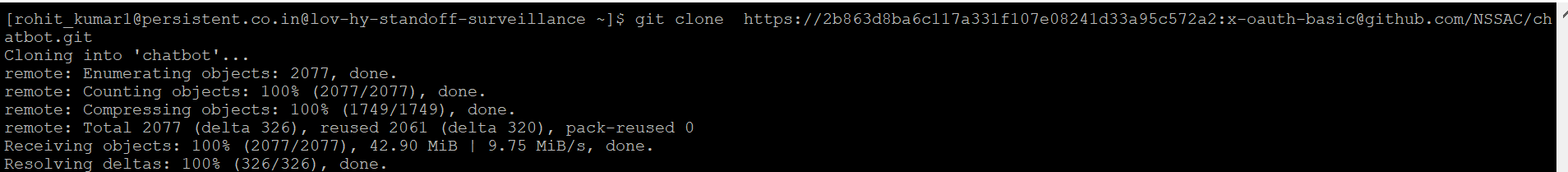


# How to get Started

This section consists of the steps need to do in order to create compose network and how to up both servers and setup intercommunication between them.

## 3.1 Clone the Repository

Clone the repository from the GitHub: git clone <https://2b863d8ba6c117a331f107e08241d33a95c572a2:x-oauth-basic@github.com/NSSAC/chatbot.git>



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:

Text

Description automatically generated

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:

A screenshot of a computer screen

Description automatically generated

## 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 in the docker-compose.yml file. You can see the images by running **docker images** command.
* As in below snapshot rasa/rasa-action-server is our image.

A picture containing text

Description automatically generated

* 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.

A picture containing background pattern

Description automatically generated

# Verification/Validation

## 4.1 Validate using Post API

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

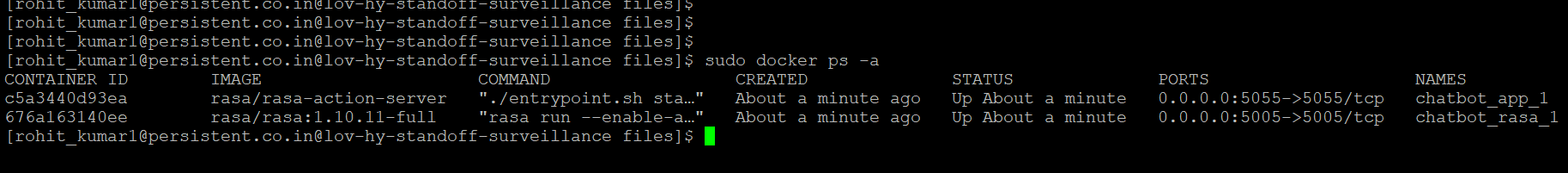
* + http://<host-ip>:5005/webhooks/rest/webhook
  + Content-type: application/json
  + Body example: {“message” : “hello”}

Graphical user interface, text, application, email

Description automatically generated

## 4.2 Verification from running containers

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



# Container actions

## 5.1 Inside container

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

**A screenshot of a computer screen

Description automatically generated**

## 5.2 chat history inside container

Chat history with chatbot is maintained in the file named listfile.txt in the container and to see the con tents or the chat history we must run cat **listfile.txt** asshown in below snapshot

A picture containing water, outdoor, people, holding

Description automatically generated

## 5.3 stopping the containers

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

# Rasa Train

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**

# Re-Deploying in case of new commits

* If there is a new commit in the GitHub, then follow below instructions:
* Stop the current 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 to know more information of 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.