CC Mini Project Report
Chatbot using Docker

Introduction:

In this project, we aim to demonstrate how cloud computing can be used to deploy an application. For the same we have demonstrated a chatbot application. Chatbot in this context refers to a simple conversational chatting bot.

Problem definition:

We have split our project into three modules:

1.Server
2.Chatbot application
3.Client

The server is a docker image hosting the model. We have used Flask to create an application for providing an interface between docker and python.

The client in our example is a chatbot application which is used as a virtual conversational partner. This application is hosted on a docker repository.

Module Description:

1. Chatbot: The chatbot is a python application which is trained using AIML files which is then stored in “.brn” file. AIML stands for Artificial Intelligence Modelling Language. AIML is an XML based markup language meant to create artificial intelligent applications. AIML makes it possible to create human interfaces while keeping the implementation simple to program, easy to understand and highly maintainable. This tutorial will teach you the basics of AIML. All the basic components of AIML with suitable examples have been discussed in this tutorial.

2. Docker repository: The docker repository is used to host the chatbot application. Docker is a free service and requires to create an account to access its services. After creating an account we pushed the respective chatbot application files on to the docker repository and then the client is supposed to pull it from the cloud in order to run it. . A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, settings. Available for both Linux and Windows based apps, containerized software will always run the same, regardless of the environment. Containers isolate software from its surroundings, for example differences between development and staging environments and help

reduce conflicts between teams running different software on the same infrastructure.

3. UI: We have used HTML for creating the user interface. It is a web based application hosted on the localhost.

Environment Required: .Python

Code:

main_raw.py

```
import aiml
import os
kernel = aiml.Kernel()
if os.path.isfile("bot_brain.brn"):
    kernel.bootstrap(brainFile = "bot_brain.brn")
else:
    kernel.bootstrap(learnFiles = os.path.abspath("aiml/std-startup.xml"), commands = "load aiml b")
    kernel.saveBrain("bot_brain.brn")
# kernel now ready for use
while True:
    message = raw_input("Enter your message to the bot: ")
    if message == "quit":
        exit()
    elif message == "save":
        kernel.saveBrain("bot_brain.brn")
    else:
        bot_response = kernel.respond(message)
        print bot_response
```

main.py

```
from flask import Flask, render_template, request, jsonify
import aiml
import os
app = Flask(__name__)
@app.route("/")
def hello():
    return render_template('chat.html')
@app.route("/ask", methods=['POST'])
def ask():
            message = str(request.form['messageText'])
            kernel = aiml.Kernel()
            if os.path.isfile("bot_brain.brn"):
                kernel.bootstrap(brainFile = "bot_brain.brn")
```

```
        else:
            kernel.bootstrap(learnFiles = os.path.abspath("aiml/std-startup.xml"), commands = "load aiml b")
            kernel.saveBrain("bot_brain.brn")
        # kernel now ready for use
        while True:
            if message == "quit":
                exit()
            elif message == "save":
                kernel.saveBrain("bot_brain.brn")
            else:
                bot_response = kernel.respond(message)
                # print bot_response
                return jsonify({'status':'OK','answer':bot_response})
if __name__ == "__main__":
    app.run(host='0.0.0.0', debug=True)
```

DockerFile

```
# our base image
FROM alpine:3.5
# Install python and pip
RUN apk add --update py2-pip
RUN mkdir -p /usr/src/app
WORKDIR /usr/src/app
# install Python modules needed by the Python app
COPY requirements.txt /usr/src/app/
RUN pip install --no-cache-dir -r /usr/src/app/requirements.txt
# copy files required for the app to run
COPY main.py /usr/src/app/
COPY main-raw.py /usr/src/app/
COPY templates/chat.html /usr/src/app/templates/chat.html
COPY aiml /usr/src/app/aiml
# tell the port number the container should expose
#EXPOSE 5000
# run the application
CMD ["python", "/usr/src/app/main.py"]
```

chat.html

```
<!DOCTYPE html>
<html>
    <head lang="en">
        <meta charset="UTF-8">
            <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Chatbot</title>
            <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
    <link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"
rel="stylesheet">
    <link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.6.3/css/font-awesome.min.css"
rel="stylesheet">
    <style type="text/css">
     .fixed-panel {
        min-height: 400px;
        max-height: 400px;
             background-color: #19313c;
             color: white;
        overflow: auto;
      }
      .media-list {
             overflow: auto;
                    clear: both;
                    display: table;
                    overflow-wrap: break-word;
                    word-wrap: break-word;
                    word-break: normal;
                    line-break: strict;
     }
                    .panel {
                    margin-bottom: 20px;
                    background-color: #fff;
                    border: 6px solid transparent;
                    border-radius: 25px;
                    -webkit-box-shadow: 0 1px 1px rgba(0,0,0,.05);
                    box-shadow: 0 1px 1px rgba(0,0,0,.05);
                    }
                    .panel-info {
                    border-color: #0c2735;
                    }
                    .panel-info>.panel-heading {
                    color: #fff;
                    background-color: #0c2735;
                    border-color: #0c2735;
                    }
                    .panel-footer {
                    padding: 10px 15px;
                    background-color: #0c2735;
                    border-top: 1px solid #0c2735;
                    border-bottom-right-radius: 3px;
                    border-bottom-left-radius: 3px;
                    }
                    body {
                    /* Permalink - use to edit and share this gradient:
http://colorzilla.com/gradient-editor/#608f95+0,008588+9,0c2449+52,1a1e3b+100 */
                    background: rgb(96,143,149); /* Old browsers */
```

```
                                        background: -moz-linear-gradient(-45deg, rgba(96,143,149,1) 0%,
rgba(0,133,136,1) 9%, rgba(12,36,73,1) 52%, rgba(26,30,59,1) 100%); /* FF3.6-15 */
                                        background: -webkit-linear-gradient(-45deg, rgba(96,143,149,1)
0%,rgba(0,133,136,1) 9%,rgba(12,36,73,1) 52%,rgba(26,30,59,1) 100%); /* Chrome10-25,Safari5.1-6 */
                                        background: linear-gradient(135deg, rgba(96,143,149,1) 0%,rgba(0,133,136,1)
9%,rgba(12,36,73,1) 52%,rgba(26,30,59,1) 100%); /* W3C, IE10+, FF16+, Chrome26+, Opera12+,
Safari7+ */
                                        filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#608f95',
endColorstr='#1a1e3b',GradientType=1 ); /* IE6-9 fallback on horizontal gradient */
                                        }
        </style>
    </head>
<body>
        <div class="container background-color: rgb(255,0,255);">
            <div class="row">
                <h3 class="text-center"><small><strong>Tanvi</strong></small><font color="white">
Talks!!! </font><small><strong>Group 3</strong></small><font color="white"> rocks..</font></h3>

                <div class="col-md-4 col-md-offset-4">

                    <div id="chatPanel" class="panel panel-info">
                        <div class="panel-heading">
                                                                        <strong><span class="glyphicon
glyphicon-globe"></span> Talk with Me !!!</strong>

                                                                </div>
                        <div class="panel-body fixed-panel">
                            <ul class="media-list">
                            </ul>
                        </div>
                        <div class="panel-footer">
                            <form method="post" id="chatbot-form">
                                <div class="input-group">
                                    <input type="text" class="form-control" placeholder="Enter Message"
name="messageText" id="messageText" autofocus/>
                                    <span class="input-group-btn">
                                        <button class="btn btn-info" type="button" id="chatbot-form-btn">SEND <span
class="glyphicon glyphicon-hand-up"></span></button>
                                    </span>
                                </div>
                            </form>
                        </div>
                    </div>
                </div>
            </div>
        </div>

        <script src="http://code.jquery.com/jquery-1.12.4.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
    <script>
    $(function() {
      $('#chatbot-form-btn').click(function(e) {
        e.preventDefault();
        $('#chatbot-form').submit();
      });

      $('#chatbot-form').submit(function(e) {
        e.preventDefault();

        var message = $('#messageText').val();
        $(".media-list").append('<li class="media"><div class="media-body"><div
class="media"><div class="media-body">' + message + '<hr/></div></div></div></li>');
        $.ajax({
          type: "POST",
          url: "/ask",
          data: $(this).serialize(),
          success: function(response) {
            $('#messageText').val('');

            var answer = response.answer;
            const chatPanel = document.getElementById("chatPanel");
            $(".media-list").append('<li class="media"><div class="media-body"><div
class="media"><div class="media-body">' + answer + '<hr/></div></div></div></li>');
                    $(".fixed-panel").stop().animate({ scrollTop: $(".fixed-panel")[0].scrollHeight},
1000);

          },
          error: function(error) {
            console.log(error);
          }
        });
      });
    });
    </script>
  </body>
</html>
```

requirements.txt

```
Flask==0.12
aiml==0.8.6
```

Build the image

```
docker build -t sanjanamoghe/chatbot
```

Run the image

```
docker run --rm -p 5000:5000 --name chatbot sanjanamoghe/chatbot
```

Pushing the image on Docker

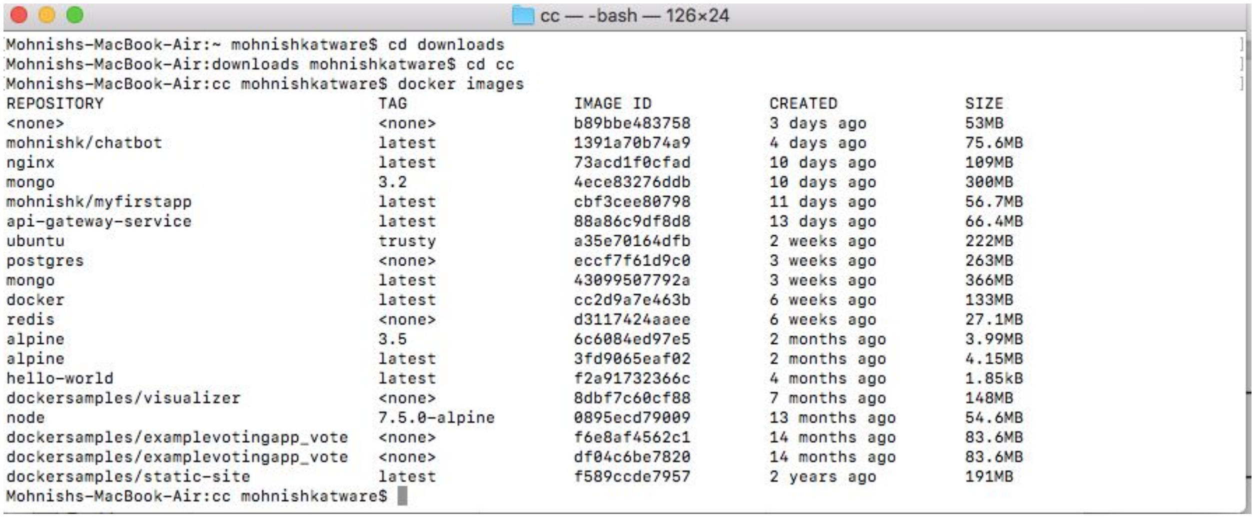
Pulling from Docker

```
docker run --rm -p 5000:5000 --name chatbot sanjanamoghe/chatbot
```

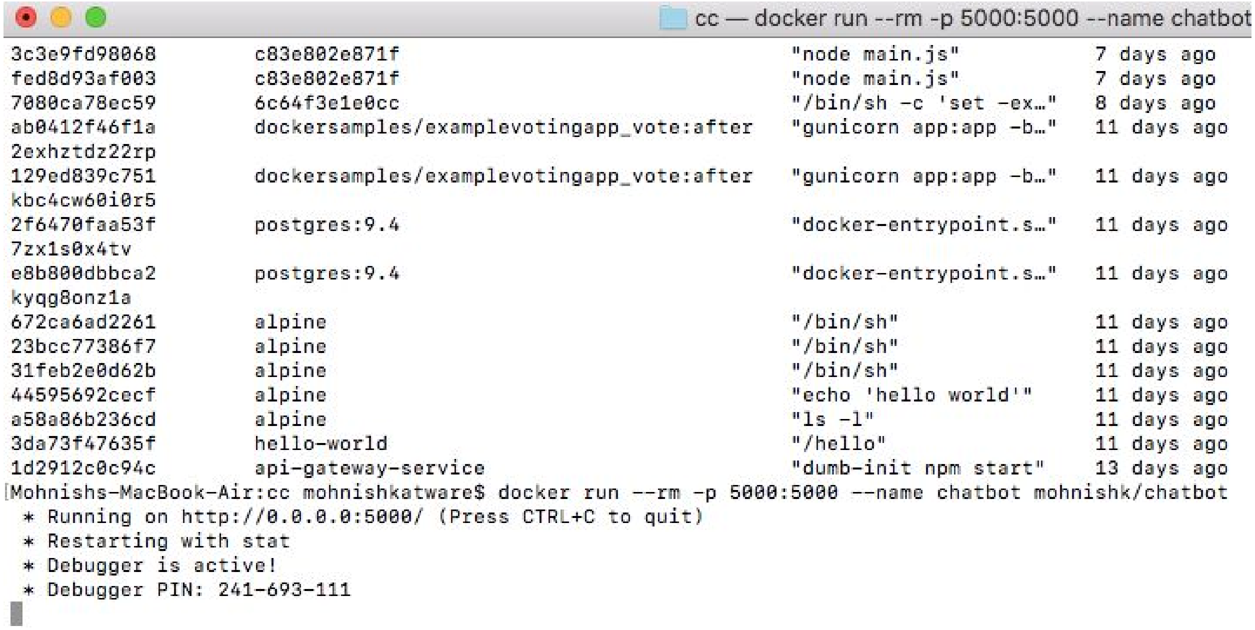
Run

```
docker run --rm -p 5000:5000 --name chatbot sanjanamoghe/chatbot
```

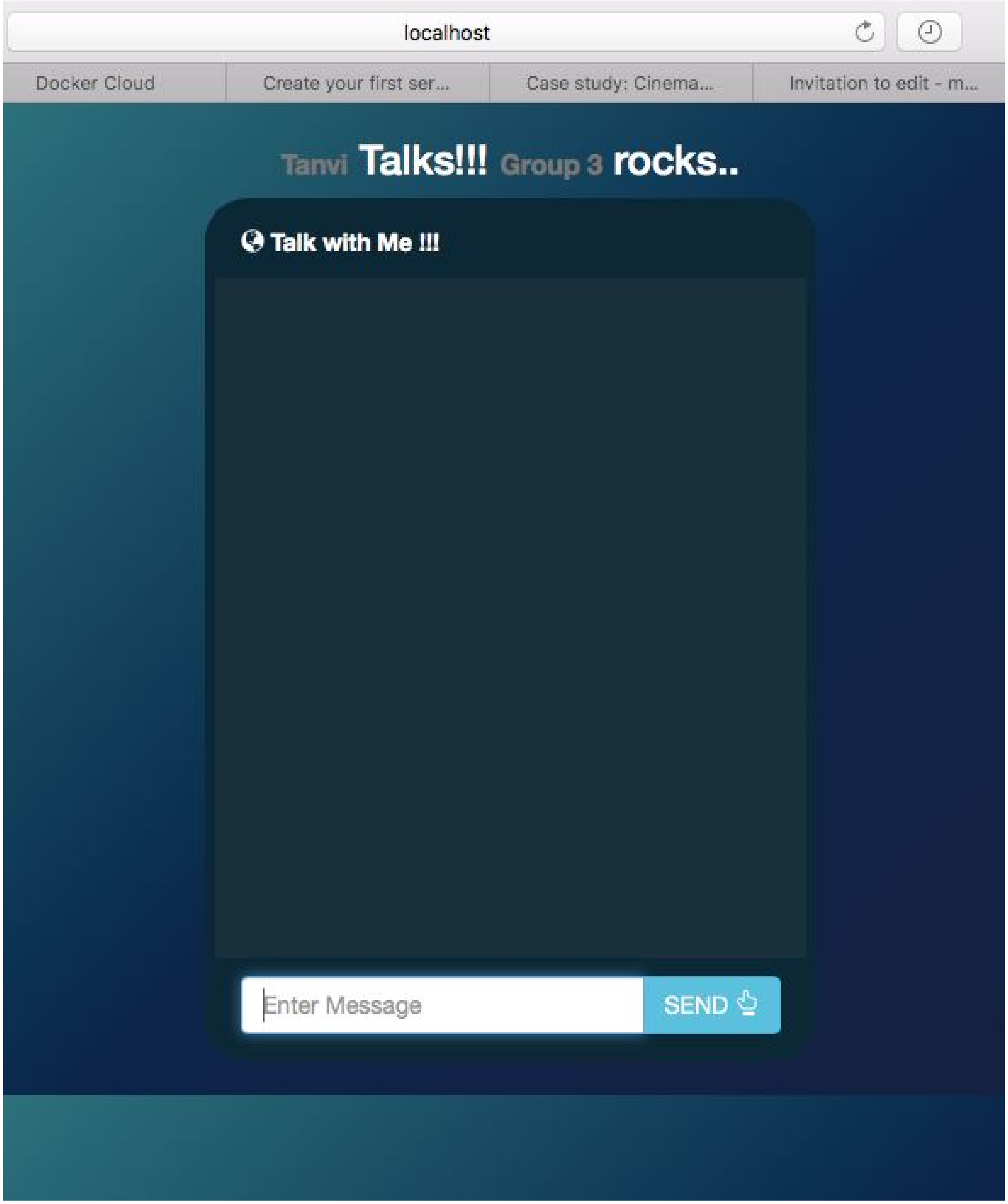
Output screenshots:



```
Mohnishs-MacBook-Air:~ mohnishkatware$ cd downloads
Mohnishs-MacBook-Air:downloads mohnishkatware$ cd cc
Mohnishs-MacBook-Air:cc mohnishkatware$ docker images
REPOSITORY                              TAG                 IMAGE ID            CREATED             SIZE
<none>                                  <none>              b89bbe483758        3 days ago          53MB
mohnishk/chatbot                        latest              1391a70b74a9        4 days ago          75.6MB
nginx                                   latest              73acd1f0cfad        10 days ago         109MB
mongo                                   3.2                 4ece83276ddb        10 days ago         300MB
mohnishk/myfirstapp                     latest              cbf3cee80798        11 days ago         56.7MB
api-gateway-service                     latest              88a86c9df8d8        13 days ago         66.4MB
ubuntu                                  trusty              a35e70164dfb        2 weeks ago         222MB
postgres                                <none>              eccf7f61d9c0        3 weeks ago         263MB
mongo                                   latest              43099507792a        3 weeks ago         366MB
docker                                  latest              cc2d9a7e463b        6 weeks ago         133MB
redis                                   <none>              d3117424aaee        6 weeks ago         27.1MB
alpine                                  3.5                 6c6084ed97e5        2 months ago        3.99MB
alpine                                  latest              3fd9065eaf02        2 months ago        4.15MB
hello-world                             latest              f2a91732366c        4 months ago        1.85kB
dockersamples/visualizer                <none>              8dbf7c60cf88        7 months ago        148MB
node                                    7.5.0-alpine        0895ecd79009        13 months ago       54.6MB
dockersamples/examplevotingapp_vote     <none>              f6e8af4562c1        14 months ago       83.6MB
dockersamples/examplevotingapp_vote     <none>              df04c6be7820        14 months ago       83.6MB
dockersamples/static-site               latest              f589ccde7957        2 years ago         191MB
Mohnishs-MacBook-Air:cc mohnishkatware$
```

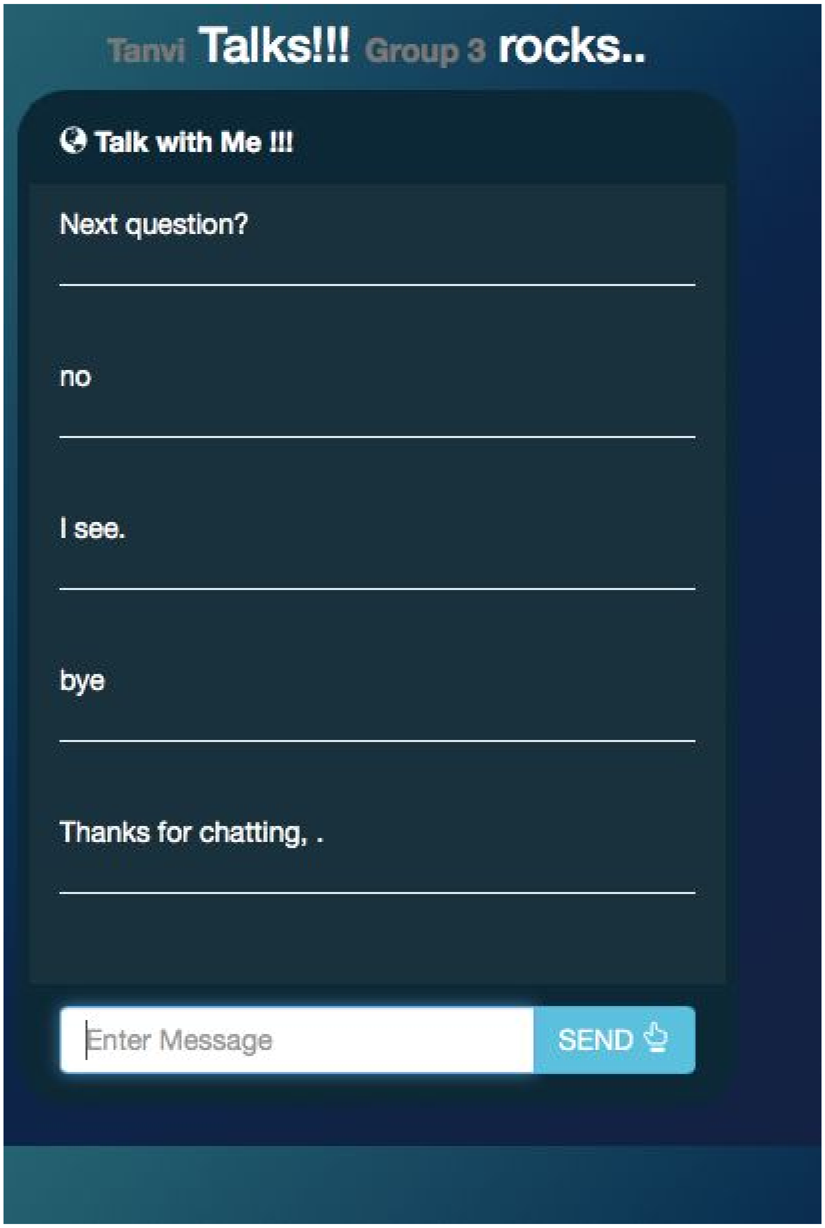


```
3c3e9fd98068        c83e802e871f                                "node main.js"            7 days ago
fed8d93af003        c83e802e871f                                "node main.js"            7 days ago
7080ca78ec59        6c64f3e1e0cc                                "/bin/sh -c 'set -ex…"    8 days ago
ab0412f46f1a        dockersamples/examplevotingapp_vote:after   "gunicorn app:app -b…"    11 days ago
2exhztdz22rp
129ed839c751        dockersamples/examplevotingapp_vote:after   "gunicorn app:app -b…"    11 days ago
kbc4cw60i0r5
2f6470faa53f        postgres:9.4                                "docker-entrypoint.s…"    11 days ago
7zx1s0x4tv
e8b800dbbca2        postgres:9.4                                "docker-entrypoint.s…"    11 days ago
kyqg8onz1a
672ca6ad2261        alpine                                      "/bin/sh"                 11 days ago
23bcc77386f7        alpine                                      "/bin/sh"                 11 days ago
31feb2e0d62b        alpine                                      "/bin/sh"                 11 days ago
44595692cecf        alpine                                      "echo 'hello world'"      11 days ago
a58a86b236cd        alpine                                      "ls -l"                   11 days ago
3da73f47635f        hello-world                                 "/hello"                  11 days ago
1d2912c0c94c        api-gateway-service                         "dumb-init npm start"     13 days ago
[Mohnishs-MacBook-Air:cc mohnishkatware$ docker run --rm -p 5000:5000 --name chatbot mohnishk/chatbot
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 241-693-111
```



Tanvi Talks!!! Group 3 rocks..

Talk with Me !!!



Next question?

no

I see.

bye

Thanks for chatting, .