VOICE BASED HOME AUTOMATION

Industrial/Practical Training Report

Submitted to the

Department of Electronics and Communication Engineering,

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE

In partial fulfillment of the requirements,

for the award of the Degree of

Bachelor of Technology
in
Electronics and Communication Engineering

By

MOHAMMAD NAZMA SULTANA

(20485A0413)

Department of Electronics and Communication Engineering

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE

SESHADRI RAO KNOWLEDGE VILLAGE

GUDLAVALLERU – 521356

ANDHRA PRADESH

2022-23

DECLARATION

I am **MOHAMMAD NAZMA SULTANA** hereby declare that this industrial training report is the record of authentic work carried out by me during the period 26.03.2022 to 30.07.2022 in **Smart Bridge Pvt.Ltd** under the supervision of my training In charge Nikhil Gurram Smart Bridge Pvt.Ltd.

Signature:

Name of the student: **MOHAMMAD NAZMA SULTANA**

Acknowledgement

I am very glad to express my deep indebtedness to all the employees of Smart Bridge Pvt.Ltd. Without their support and guidance, it would not have been possible for this training to have materialized and taken a concrete shape. I extend my deep gratitude to my training mentor – Nikhil Gurram, Gnaneswar Bandari, Sai Rajeti, who supported us all the time. I owe my personal thanks to our HOD **Dr.Y.Rama Krishna** and our industrial training coordinator **Mr. N.Samba Murthy** for undergoing training at a reputed company like Smart Bridge Pvt.Ltd.

We would like to take this opportunity to express our profound sense of gratitude to our beloved Principal **Dr.G.V.S.N.R.V.Prasad**, for providing us all the required facilities. We express our thanks to the Teaching and Non-Teaching staff of the electronics and communication engineering department who helped us directly or indirectly in completion of our internship.

MOHAMMAD NAZMA SULTANA

20485A0413

CONTENTS

TITLE

ABSTRACT

We have done our internship at **Smart Bridge** in Virtual mode. There are different areas in Electronics and communication Engineering and we have chosen Internet of Things (IoT), a newly emerged technology with a group of students.

In our Internship, we have studied various Virtual Platforms like Tinker cad, IBM Cloud Platform and Python Programming. We have learnt how to use Virtual Environment to perform the Hardware Projects using the Tinker cad and ARDUINO. We have done the IoT based Smart Irrigation System.

As an application of these, we have done a project consisting of a Team of 4 members. Our Project name is IoT Analytics in Energy Management. With the help of Smart Internz Platform we started our project with the knowledge we have gained in this Internship. We successfully completed our Project using Python Programming and IBM Cloud Services.

In this way the knowledge we have gained in the Smart Bridge company is useful to us.

Key words:

- ✓ IoT
- ✓ Arduino
- ✓ cloud
- ✓ virtual.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 01

1.0 Introductions to IoT:

What is IoT?

● Connecting everyday things embedded with electronics, software and sensors to the internet enabling them to collect and exchange data.

Benefits of IoT:

1. Improved Performance
2. Reduced Cost
3. Improved Data Collection
4. Improved Customer Engagement
5. New Revenue Streams

Applications:

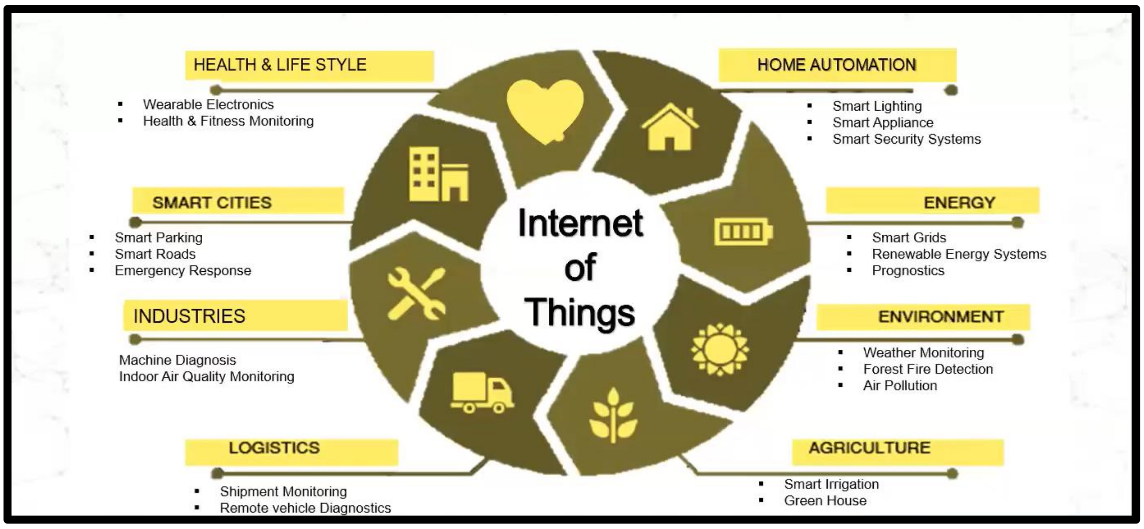


Fig 1: Internet of Things (IoT) Application in different Sectors

What is IoT Architecture?

IoT architecture is the system of numerous elements: sensors, protocols, actuators, cloud services, and layers. Given its complexity, there exist 4 stages of IoT architecture. Such a number is chosen to steadily include these various types of components into a sophisticated and unified network.

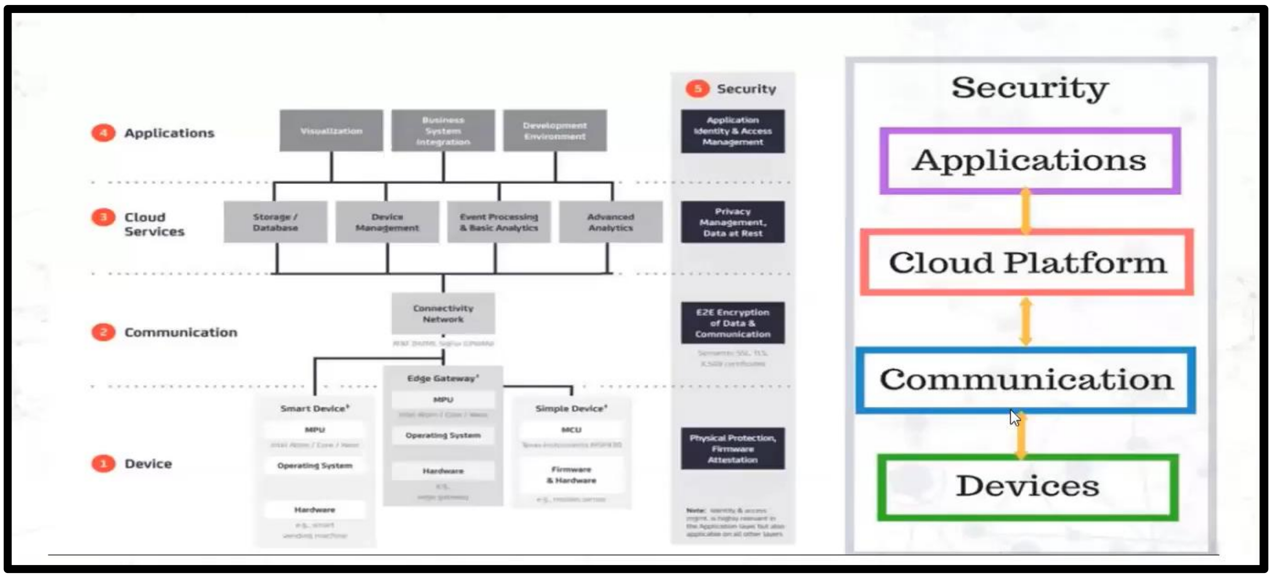


Figure 2: IoT Architecture

IoT Technology Stack:

● The IoT technology stack is nothing else than a range of technologies, standards and applications, which lead from the simple connection of objects to the Internet to the most easy and most complex applications that use these connected things, the data they gather and communicate and the different steps needed to power these applications.

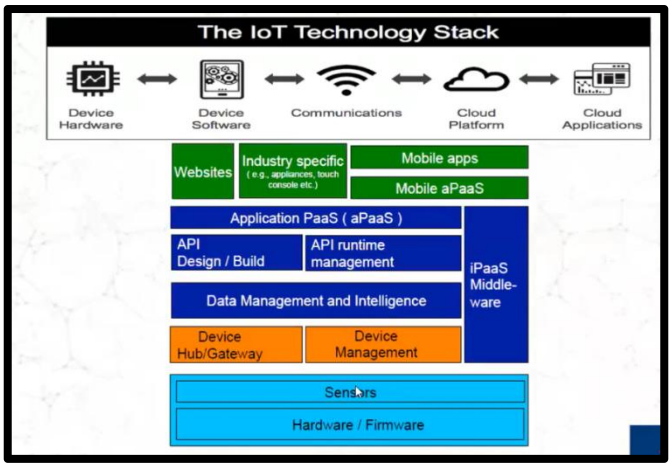


Figure 3: IoT Technology Stack

1.1 Introduction to open Hardware Platforms and Tinker cad Circuits:

What is Arduino?

- Arduino is an open-source platform used for building electronics projects.
- Easy tool for fast prototyping.
- Consists of both a physical programmable circuit board and a piece of software.

Why Arduino?

- Open-Source Platform
- Inexpensive
- Does not need a separate piece of hardware
- Arduino IDE uses a simplified version of C++
- Cross-Platform
- Provides a standard form factor.

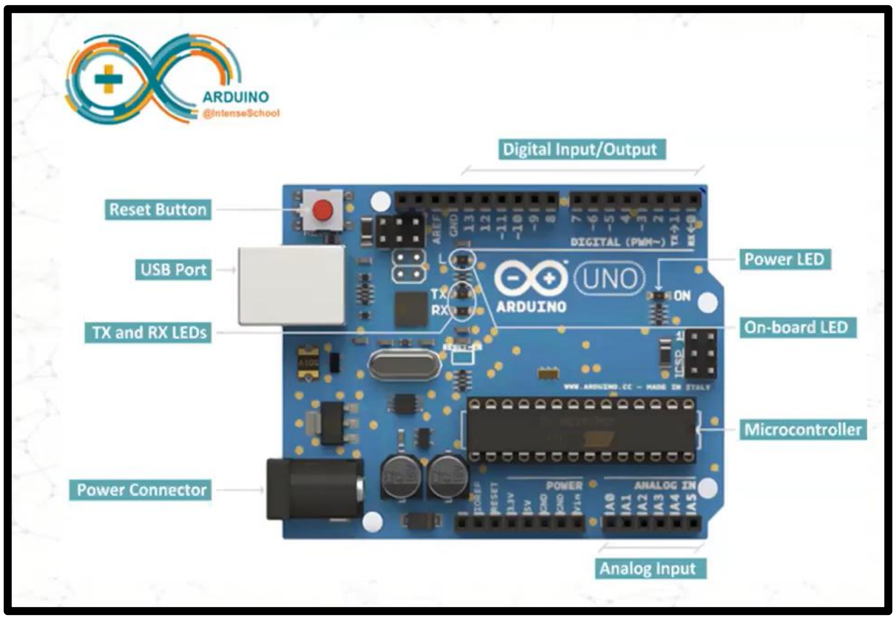


Figure 4: Arduino UNO Board

Arduino UNO Features:

Operating Voltage	5V and 3.3V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3 V Pin	50 mA
Flash Memory	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
Microcontroller	ATmega328

Table 1: Arduino UNO Feature

1.2 Arduino – Tinkercad:

Use of Tinkercad

- Tinkercad is a free, easy-to-use app for 3D design, electronics, and coding. It's used by teachers, kids, hobbyists, and designers to imagine, design, and make anything!

List of the Experiments done with Thinkercad:

- Buzz
- Smart Door
- Blinker with TMP
- PIR Sensor
- LED Blink
- Piezo
- Ultrasonic Sensor
- Temp Sensor
- Servo Motor
- LED with Potentiometer
- Digital Input and Output

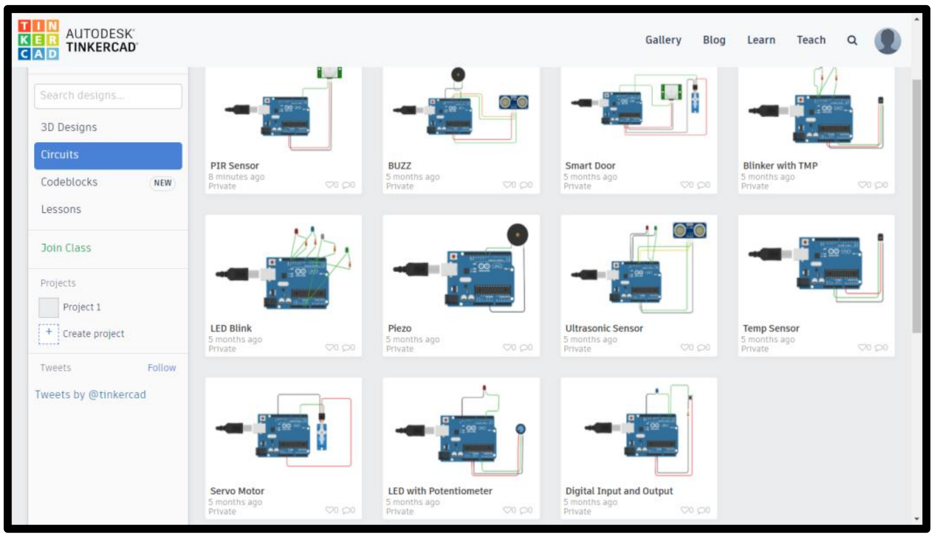


Figure 5: Experiments in Tinkercad

1.3 Introduction to Python and Python Basics:

1.3.0 Python Introduction:

What is Python?

- General Purpose
- Interpreted
- High Level
- Dynamically Typed

General Purpose:
Python language is a general-purpose language because it can be applicable to all domains, such type of languages is generic and not specialized.

Interpreted:
An Interpreted language will read the Raw code without being explicitly compiled before execution.

High Level:
Python allows programmers to express their logic in a form which is very close to human language. This helps in many computing aspects like memory management and data management.

Dynamically Typed:
Python identifies the type of variables on the basis of what kind of data you have assigned to the variable.

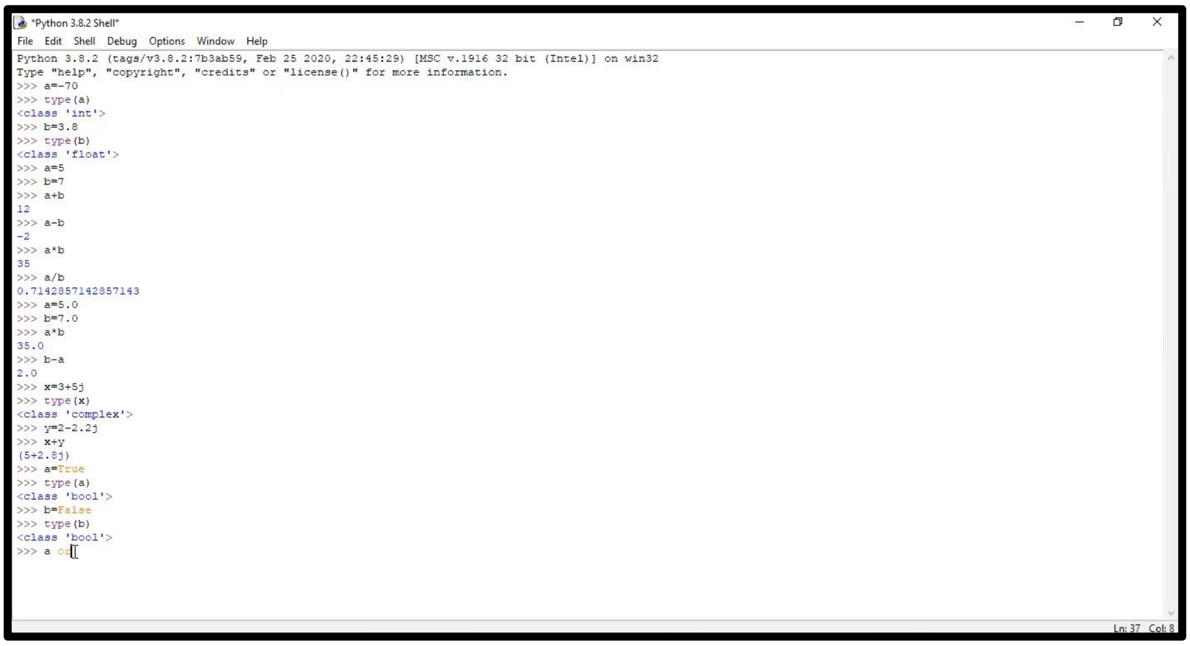
Why Choose Python?
- Readily available and open source
- Huge Libraries
- Easy to understand and learn
- Big developer Communities
- Fewer code lines and larger Functionalities.

1.3.1 Python Basics:

Here is a list of the Python keywords. Enter any keyword to get more help.

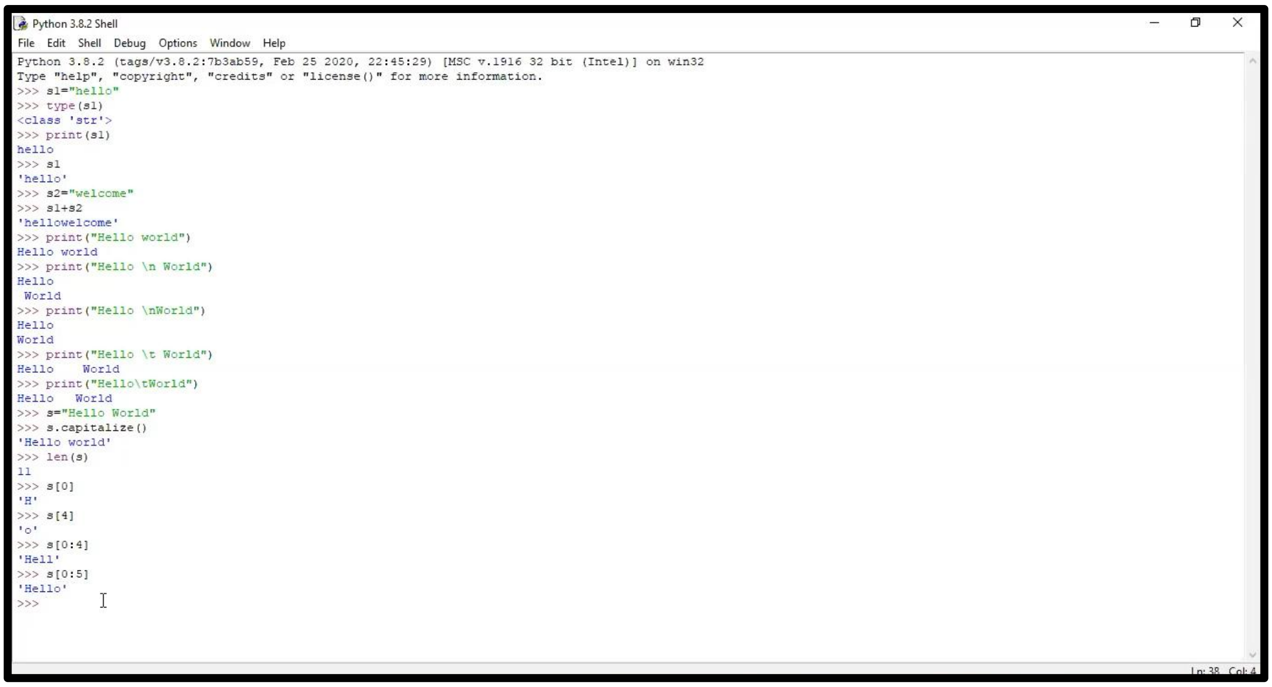
False	class	from	or
None	continue	global	pass
True	def	if	raise
and	del	import	return
as	elif	in	try
assert	else	is	while
async	except	lambda	with
await	finally	nonlocal	yield
break	for	not	

Basic Python Programming: -



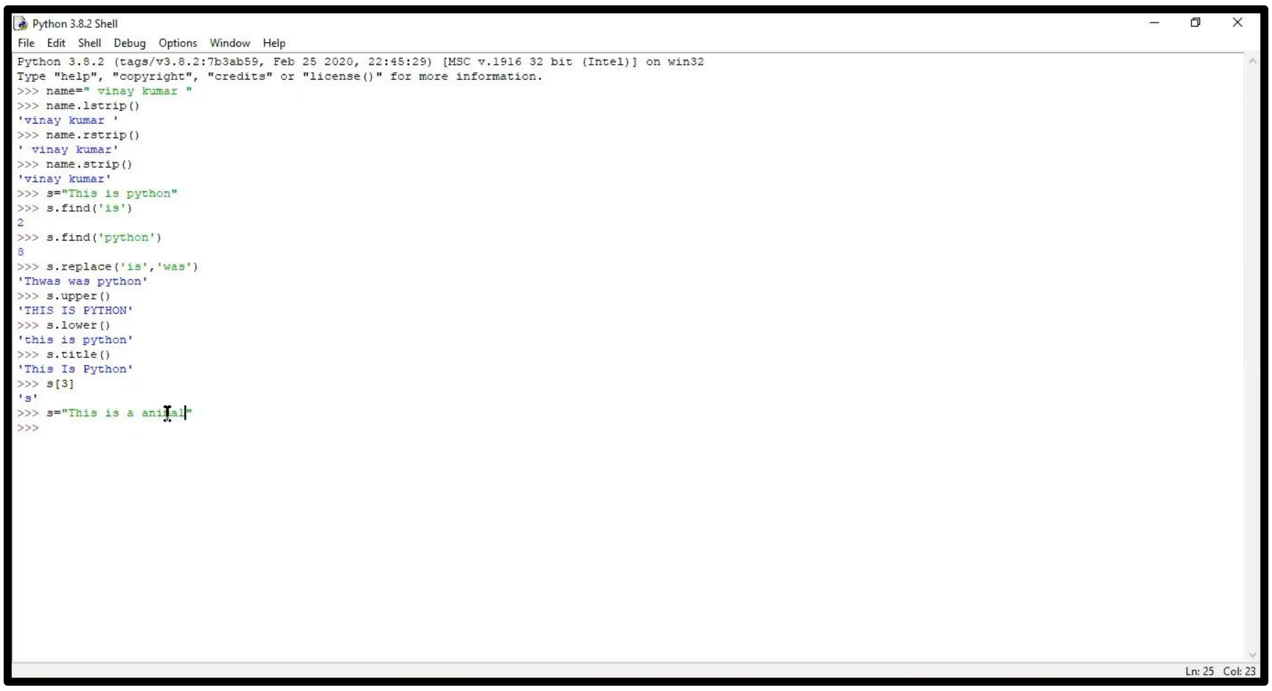
```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> a=-70
>>> type(a)
<class 'int'>
>>> b=3.8
>>> type(b)
<class 'float'>
>>> a=5
>>> b=7
>>> a+b
12
>>> a-b
-2
>>> a*b
35
>>> a/b
0.7142857142857143
>>> a=5.0
>>> b=7.0
>>> a*b
35.0
>>> b-a
2.0
>>> x=3+5j
>>> type(x)
<class 'complex'>
>>> y=2-2.2j
>>> x+y
(5+2.8j)
>>> a=True
>>> type(a)
<class 'bool'>
>>> b=False
>>> type(b)
<class 'bool'>
>>> a or
```

Figure 6: Addition, Subtraction, Multiplication, Division using Python.



```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> s1="hello"
>>> type(s1)
<class 'str'>
>>> print(s1)
hello
>>> s1
'hello'
>>> s2="welcome"
>>> s1+s2
'hellowelcome'
>>> print("Hello world")
Hello world
>>> print("Hello \n World")
Hello
 World
>>> print("Hello \nWorld")
Hello
World
>>> print("Hello \t World")
Hello    World
>>> print("Hello\tWorld")
Hello   World
>>> s="Hello World"
>>> s.capitalize()
'Hello world'
>>> len(s)
11
>>> s[0]
'H'
>>> s[4]
'o'
>>> s[0:4]
'Hell'
>>> s[0:5]
'Hello'
>>>
```

Figure 7: String, Hello World in Python Programming.



```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> name=" vinay kumar "
>>> name.lstrip()
'vinay kumar '
>>> name.rstrip()
' vinay kumar'
>>> name.strip()
'vinay kumar'
>>> s="This is python"
>>> s.find('is')
2
>>> s.find('python')
8
>>> s.replace('is','was')
'Thwas was python'
>>> s.upper()
'THIS IS PYTHON'
>>> s.lower()
'this is python'
>>> s.title()
'This Is Python'
>>> s[3]
's'
>>> s="This is a animal"
>>>
```

Figure 8: Various Operations using Python.

1.3.2 Conditional Statements and Functional basics:

Conditional Statements:

A conditional statement in Python is handled by if statements and we saw various other ways we can use conditional statements like if and else over here.

- "if condition" – It is used when you need to print out the result when one of the conditions is true or false.
- "else condition"- it is used when you want to print out the statement when your one condition fails to meet the requirement
- "elif condition" – It is used when you have a third possibility as the outcome. You can use multiple elif conditions to check for 4th,5th,6th possibilities in your code
- We can use minimal code to execute conditional statements by declaring all condition in single statement to run the code
- If Statement can be nested.

Syntax:

```
if expression
        Statement
else expression
        Statement
```

Basic Functions:

What is a function in Python?

- A function is a group of related statements that performs a specific task.
- Functions help break our program into smaller and modular chunks. As our program grows larger and larger, functions make it more organized and manageable.
- It avoids repetition and makes the code reusable.

Syntax:

```
Def function_name(parameters) :
        “””docstring”””
        Statements(s)
```

Function definition that consists of the following components:

1. Keyword **def** that marks the start of the function header.
2. A function name to uniquely identify the function. Function naming follows the same rules of writing identifiers in python.
3. Parameters (arguments) through which we pass values to a function. They are optional.
4. A colon (:) to mark the end of the function header.
5. Optional documentation string (docstring) to describe what the function does.
6. One or more valid python statements that make up the function body. Statements must have the same indentation level (usually 4 spaces).
7. An optional **return** statement to return a value from the function.

Types of Functions:

1. Built-in functions - Functions that are built into Python.
2. User-defined functions- Functions defined by the users themselves.

1.3.3 Data Variables and Basic Variables:

Data Types in Python:

Every value in Python has a data type. Since everything is an object in Python programming, data types are actually classes and variables are instances (object) of these classes.

There are various data types in Python. Some of the important types are listed below.

1. **Python Numbers:**

➢ Integers, floating point numbers and complex numbers fall under the Python numbers category. They are defined as int, float, and complex classes in Python.

➢ We can use the **type()** function to know which class a variable or a value belongs to. Similarly, the **isinstance()** function is used to check if an object belongs to a particular class.

2. **Python Strings:**

➢ String is a sequence of Unicode characters. We can use single quotes or double quotes to represent strings. Multi-line strings can be denoted using triple quotes, ‘’ or **“””.**

3. **Python Set:**

➢ Set is an unordered collection of unique items. Set is defined by values separated by commas inside braces **{ }.** Items in a set are not ordered.

1.3.4 **Lists, Tuples and Dictionaries:**

1. **Python List:**

● List is an ordered sequence of items. It is one of the most used data types in Python and is very flexible. All the items in a list do not need to be of the same type.

● Declaring a list is pretty straight forward. Items separated by commas are enclosed within brackets **[ ]**.

● We can use the slicing operator **[ ]** to extract an item or a range of items from a list. The index starts from 0 in Python.

2. **Python Tuple:**

● Tuple is an ordered sequence of items the same as a list. The only difference is that tuples are immutable. Tuples once created cannot be modified.
● Tuples are used to write-protect data and are usually faster than lists as they cannot change dynamically.
● It is defined within parentheses **( )** where items are separated by commas.
● We can use the slicing operator **[ ]** to extract items but we cannot change its value.

3. **Python Dictionaries:**

● Dictionary is an unordered collection of key-value pairs.
● It is generally used when we have a huge amount of data. Dictionaries are optimized for retrieving data. We must know the key to retrieve the value.
● In Python, dictionaries are defined within braces **{ }** with each item being a pair in the form **key:value**. Key and value can be of any type.
● We use keys to retrieve the respective value. But not the other way around.

1.3.5 **Functions and Modules:**

Python Functions:

● Functions are a handy way to isolate a particular part of your program's functionality and make it reusable. Modules are a way to collect a number of helpful functions in one file, which you can then use in multiple projects and share with other programmers.
● We've already been using functions extensively in our programs; functions are one of the main building blocks of Python programming. Whenever you've typed something like **len(x)** or **type(y) or even random.choice ([1, 2, 3]),** you've been using functions. It's just that these functions come pre-defined by Python.

Python Modules:

● A Python module is a file that contains one or more function definitions. Modules are a handy way of keeping related functions together, so that you can easily reuse them between projects, or share them with other programmers.
● Making a module is easy. Just make a file that has a **.py** extension and contains only **import** statements and function definitions. Here's a module called **restaurantutils** that contains many of the functions.

1.3.6 **Files IO and Python Inbuilt Libraries:**

Files:

● Files are named locations on disk to store related information. They are used to permanently store data in a non-volatile memory.
● RAM is volatile which loses its data when the computer is turned off, we use files for future use of the data by permanently storing them.
● When we want to read from or write to a file, we need to open it first. When we are done, it needs to be closed so that the resources that are tied with the file are freed.

Hence, in Python, a file operation takes place in the following order:

1. Open a file
2. Write
3. Read (perform operation)
4. Close the file

1. **Opening Files in Python:**

● Python has a built-in **open( )** function to open a file. This function returns a file object, also called a handle, as it is used to read or modify the file accordingly.

● We can specify the mode while opening a file. In mode, we specify whether we want to read **r**, write **w** or append **a** to the file. We can also specify if we want to open the file in text mode or binary mode.

Mode	Description
R	Open a file for reading. (default)
w	Open a file for writing. Creates a new file if it does not exist or truncates the file if it exists.
X	Opens a file for exclusive creation. If the file already exists, the operation fails.
A	Opens a file for appending at the end of the file without truncating it. Creates a new file if it does not exist.
T	Opens a file for appending at the end of the file without truncating it. Creates a new file if it does not exist.
B	Opens in binary mode.
+	Opens a file for updating (reading and writing)

Table 2: Modes description in Opening files in python

2. **Writing to files in Python:**

● In order to write into a file in Python, we need to open it in write **w**, append **a** or exclusive creation **x** mode.

● We need to be careful with the **w** mode, as it will overwrite into the file if it already exists.

● Writing a string or sequence of bytes for binary files is done using the **write( )** method. This method returns the number of characters written to the file.

● This program will create a new file named **test.txt** in the current directory if it does not exist. If it does exist, it is overwritten.

3. **Reading Files in Python:**

● To read a file in Python, we must open the file in reading **r** mode.

● There are various methods available for this purpose. We can use the **read(size)** method to read in the **size** number of data. If the **size** parameter is not specified, it reads and returns up to the end of the file.

● We can see that the **read( )** method returns a newline as **'\n'**. Once the end of the file is reached, we get an empty string on further reading.

● We can change our current file cursor position using the **seek( )** method. Similarly, the **tell( )** method returns our current position in the number of bytes.

4. **Closing Files in Python:**

● When we are done with performing operations on the file, we need to properly close the file.

● Closing a file will free up the resources that were tied with the file. It is done using the **close( )** method available in Python.

● Python has a garbage collector to clean up unreferenced objects but we must not rely on it to close the file.

Python File Methods:

There are various methods available with the file object. Some of them have been used in the above examples.

Here is the complete list of methods in text mode with a brief description:

Method	Description
close()	Closes an opened file. It has no effect if the file is already closed.
detach()	Separates the underlying binary buffer from the **Text0Base** and returns it.
fileno()	Returns an integer number (file descriptor) of the file.
flush()	Flushes the write buffer of the file stream.
isatty()	Returns **True** if the file stream is interactive.
read(**n**)	Reads at most **n** characters from the file. Reads till end of file if it is negative or **None.**
readable()	Returns **True** if the file stream can be read from.
readline(**n**=-1)	Reads and returns one line from the file. Reads in at most **n** bytes if specified.
readlines(**n**=-1)	Reads and returns a list of lines from the file. Reads in at most **n** bytes/characters if specified.
seek(**offset**, **from**, **SEEK_SET**)	Changes the file position to **offset** bytes, in reference to **from** (start, current, end).
seekable()	Returns **True** if the file stream supports random access.
tell()	Returns the current file location.
truncate (**size** = **None**)	Resizes the file stream to **size** bytes. If **size** is not specified, resizes to current location.
writable()	Returns **True** if the file stream can be written to.
write(**s**)	Writes the string **s** to the file and returns the number of characters written.
writelines(**lines**)	Writes a list of **lines** to the file.

Table 3: Python File Methods Description

1.3.7 **Python OOPS Concepts:**

Python Object Oriented Programming:

- Python is a multi-paradigm programming language. It supports different programming approaches.
- One of the popular approaches to solve a programming problem is by creating objects. This is known as Object-Oriented Programming (OOP)

An object has two characteristics:

- Attributes
- Behavior

In Python, the concept of OOP follows some basic principles:

1) **Class:**
- A Class is a blueprint for the object.
- We can think of class as a sketch of a parrot with labels. It contains all the details about the name, colors, size etc. Based on these descriptions, we can study about the parrot. Here, a parrot is an object.

2) **Object:**
- An object (instance) is an instantiation of a class. When class is defined, only the description for the object is defined. Therefore, no memory or storage is allocated.

3) **Methods:**
- Methods are functions defined inside the body of a class. They are used to define the behaviors of an object.

4) **Inheritance:**
- Inheritance is a way of creating a new class for using details of an existing class without modifying it. The newly formed class is a derived class (or child class). Similarly, the existing class is a base class (or parent class).

5) **Encapsulation:**
- Using OOP in Python, we can restrict access to methods and variables. This prevents data from direct modification which is called encapsulation. In Python, we denote private attributes using underscore as the prefix i.e single **_** or double **__**.

6) **Polymorphism:**
- Polymorphism is an ability (in OOP) to use a common interface for multiple forms (data types).
- Suppose, we need to color a shape, there are multiple shape options (rectangle, square, circles). However we could use the same method to color any shape. This concept is called Polymorphism.

Key Points on OOPS:

1) Object-Oriented Programming makes the program easy to understand as well as efficient.
2) Since the class is sharable, the code can be reused.
3) Data is safe and secure with data abstraction.
4) Polymorphism allows the same interface for different objects, so programmers can write code efficiently.

1.3.8 **Networking - Socket Programming:**

Networking:

A computer network is a group of computers that use a set of common communication protocols over digital interconnections for the purpose of sharing resources located on or provided by the network nodes.

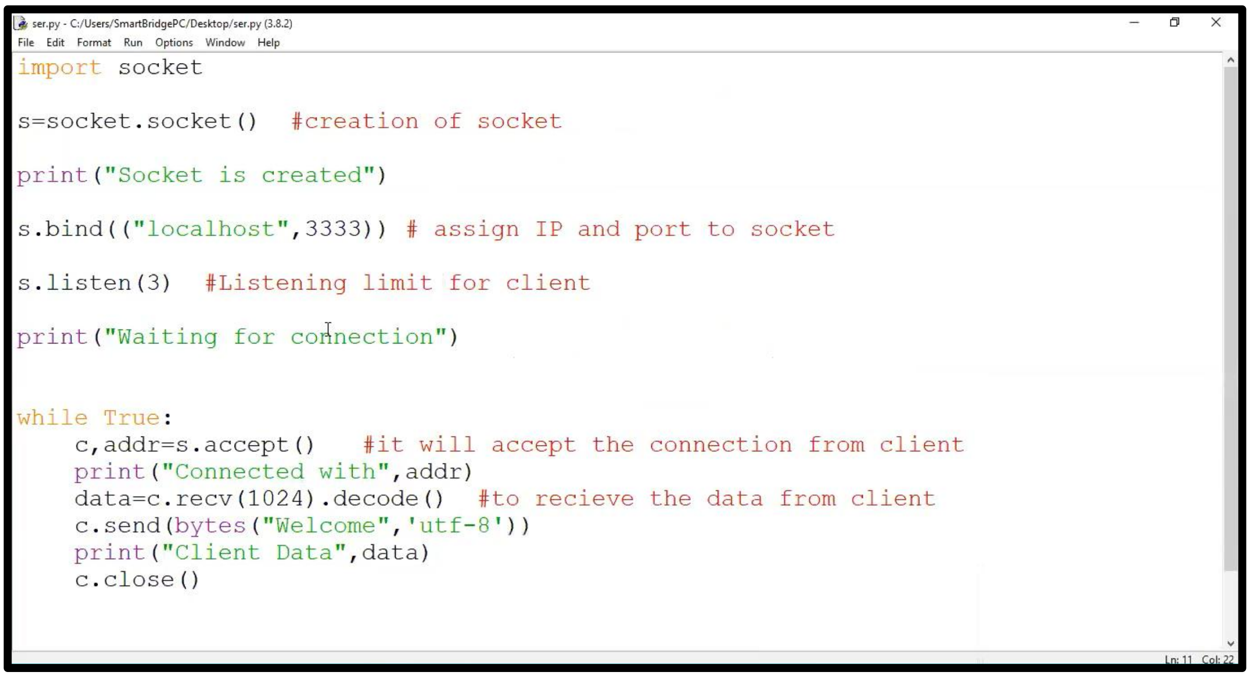
OSI Model:

1) Physical
2) Data Link
3) Network
4) Transport
5) Session
6) Presentation
7) Application.

TCP/IP & Socket Programming:

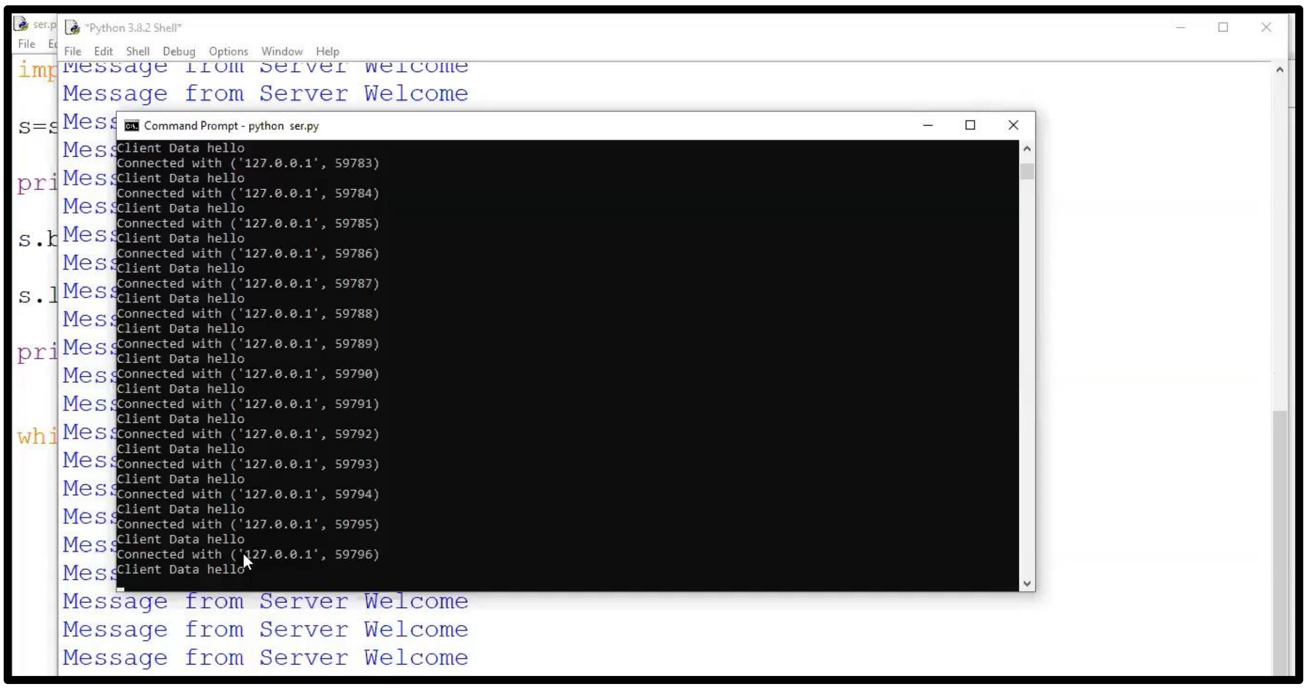
A Socket Programming interface provides the routines required for interprocess communication between applications, either on the local system or spread in a distributed, TCP/IP based network environment. Socket is uniquely defined by

- Internet Address (IP Address)
- Communication Protocol
- Port


ser.py - C:/Users/SmartBridgePC/Desktop/ser.py (3.8.2)
File Edit Format Run Options Window Help
Ln: 11 Col: 22

```
import socket

s=socket.socket()  #creation of socket

print("Socket is created")

s.bind(("localhost",3333)) # assign IP and port to socket

s.listen(3)  #Listening limit for client

print("Waiting for connection")


while True:
    c,addr=s.accept()   #it will accept the connection from client
    print("Connected with",addr)
    data=c.recv(1024).decode()  #to recieve the data from client
    c.send(bytes("Welcome",'utf-8'))
    print("Client Data",data)
    c.close()
```

Figure 9: Socket Programming Code- Creating Socket & Waiting for Connection

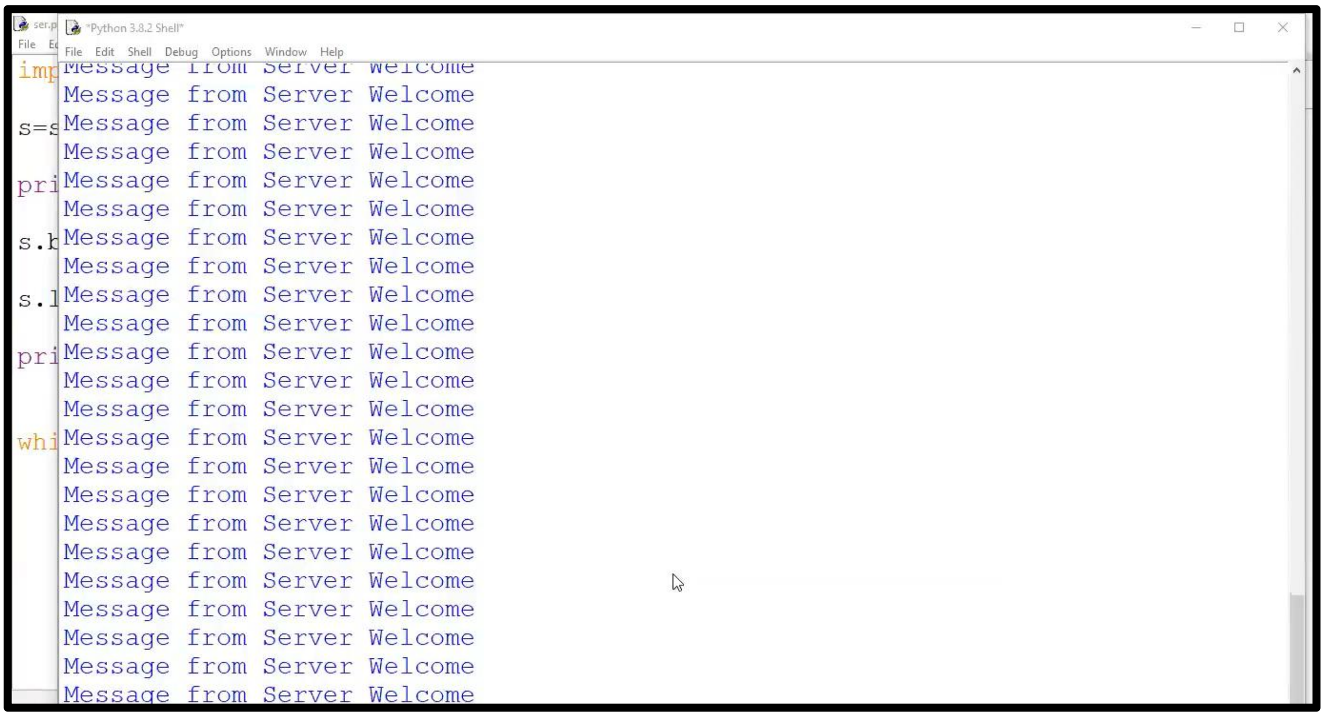

"Python 3.8.2 Shell"
File Edit Shell Debug Options Window Help
Command Prompt - python ser.py

```
Message from Server Welcome
Message from Server Welcome
```

```
Client Data hello
Connected with ('127.0.0.1', 59783)
Client Data hello
Connected with ('127.0.0.1', 59784)
Client Data hello
Connected with ('127.0.0.1', 59785)
Client Data hello
Connected with ('127.0.0.1', 59786)
Client Data hello
Connected with ('127.0.0.1', 59787)
Client Data hello
Connected with ('127.0.0.1', 59788)
Client Data hello
Connected with ('127.0.0.1', 59789)
Client Data hello
Connected with ('127.0.0.1', 59790)
Client Data hello
Connected with ('127.0.0.1', 59791)
Client Data hello
Connected with ('127.0.0.1', 59792)
Client Data hello
Connected with ('127.0.0.1', 59793)
Client Data hello
Connected with ('127.0.0.1', 59794)
Client Data hello
Connected with ('127.0.0.1', 59795)
Client Data hello
Connected with ('127.0.0.1', 59796)
Client Data hello
```

```
Message from Server Welcome
Message from Server Welcome
Message from Server Welcome
```

Figure 10: Command Prompt Output-Client Data



```
Message from Server Welcome
Message from Server Welcome
Message from Server Welcome
Message from Server Welcome
Message from Server Welcome
Message from Server Welcome
Message from Server Welcome
Message from Server Welcome
Message from Server Welcome
Message from Server Welcome
Message from Server Welcome
Message from Server Welcome
Message from Server Welcome
Message from Server Welcome
Message from Server Welcome
Message from Server Welcome
Message from Server Welcome
Message from Server Welcome
Message from Server Welcome
Message from Server Welcome
Message from Server Welcome
Message from Server Welcome
```

Figure 11: Output from Python Compiler - From Server.

CHAPTER 02

2.1 IBM Cloud Platforms:

Cloud:

Cloud Computing is simply on-demanded delivery of computing services over the internet.

Cloud Computing is categorized into different service models:

- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Infrastructure as a Service (IaaS)

Software as a Service (SaaS):

- Simplest to use and easier to deploy
- No software to install or configure
- Everything is available by web browser.
- Microsoft's office and slack are two popular SaaS products.

Platform as a Service (PaaS):

- Developers and programmers upload their content to pre-configured servers.
- No installation or maintaining the server software or operating system.

Infrastructure as a Service (IaaS):

- Dedicated virtual systems that you manage and maintain yourself.
- Greatest degree of flexibility and customization in cloud computing.
- Amazon web services (AWS) and Digital ocean are widely known as IaaS providers.

IBM IoT Platform Architecture:

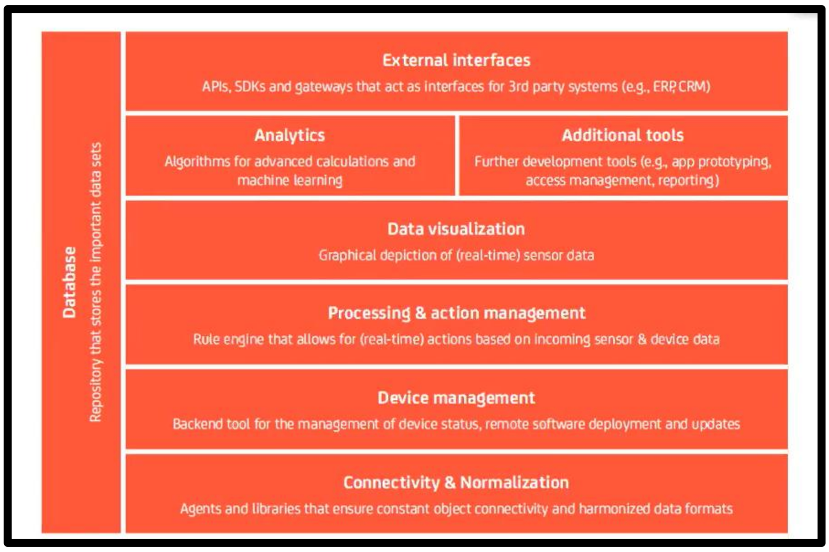

Database
Repository that stores the important data sets
External interfaces
APIs, SDKs and gateways that act as interfaces for 3rd party systems (e.g., ERP, CRM)
Analytics
Algorithms for advanced calculations and machine learning
Additional tools
Further development tools (e.g., app prototyping, access management, reporting)
Data visualization
Graphical depiction of (real-time) sensor data
Processing & action management
Rule engine that allows for (real-time) actions based on incoming sensor & device data
Device management
Backend tool for the management of device status, remote software deployment and updates
Connectivity & Normalization
Agents and libraries that ensure constant object connectivity and harmonized data formats

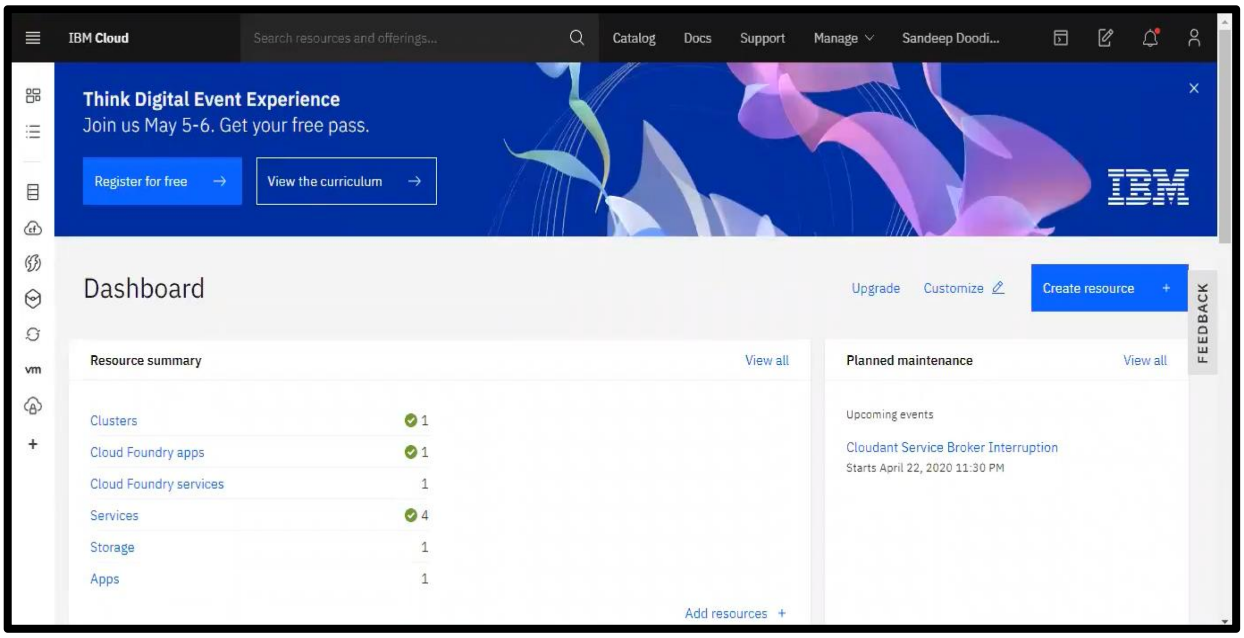
Figure 12: IBM IoT Platform Architecture

1. Connectivity & Normalization: Brings different protocols and different data formats into one “software” interface ensuring accurate data streaming and interaction with all devices.
2. Device Management: Ensures the connected “things” are working properly, seamlessly running patches and updates for software and applications running on the device or edge gateways.
3. Database: Scalable storage of device data brings the requirements for hybrid cloud-based databases to a new level in terms of data volume, variety, velocity and veracity.
4. Processing & Action Management: Brings data to life with rule-based event-action-triggers enabling execution of “smart” actions based on specific sensor data.
5. Analytics: Performs a range of complex analysis from basic data clustering and deep machine learning to predictive analytics extracting the most value out of the IoT data-stream.
6. Visualization: Enables humans to see patterns and observe trends from visualization dashboards where data is vividly portrayed through line-, stacked-, or pie charts, 2D- or even 3D-models.
7. Additional Tools: Allow IoT developers prototype, test and market the IoT use case creating platform ecosystem apps for visualizing, managing and controlling connected devices.
8. External Interfaces: Integrate with 3rd-party systems and the rest of the wider IT-ecosystem via built-In application programming interfaces (API), software development kits (SDK), and gateways.

2.1.1 **IBM IoT Device creation and connecting Internal Simulator:**

Steps for IBM IoT device creation and connecting Internal Simulator:

1. First we have to Create an IBM Cloud Account at **https://www.ibm.com/in-en/cloud** **.**
2. Next we have to login into our account.
3. You will display the IBM dashboard. In that dashboard you can see our active IBM Services.
4. Click on the catalog we will get the list of the services provided by the IBM Cloud.
5. Create Internet of Things Service.
6. After service is Created, we have to add devices to the IoT Platform.
7. We can add devices like Raspberry Pie, Arduino etc.
8. We have to connect the device to the IBM virtual IoT simulator http://watson-iot-sensor-simulator.mybluemix.net/ .



Dashboard

Resource summary · View all

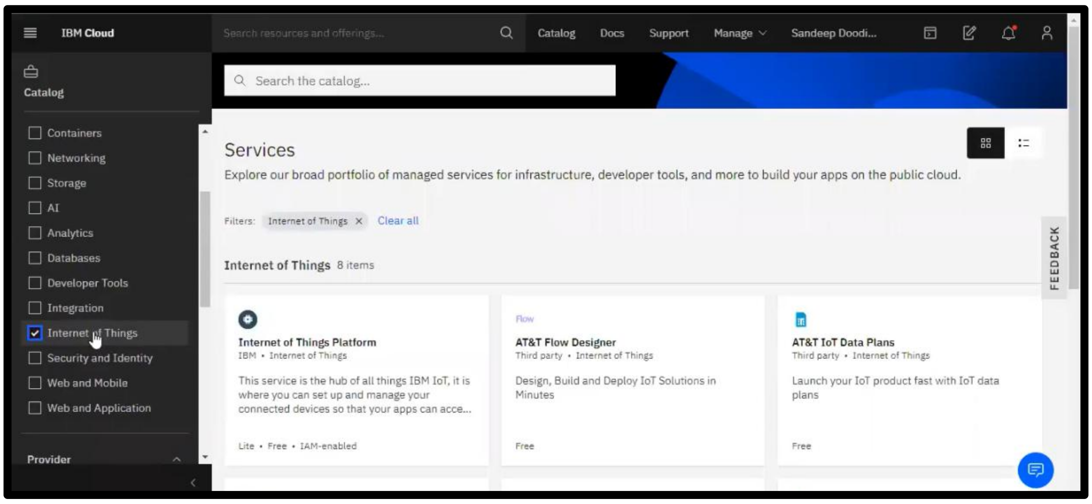
Clusters	✓ 1
Cloud Foundry apps	✓ 1
Cloud Foundry services	1
Services	✓ 4
Storage	1
Apps	1

Add resources +

Planned maintenance · View all

Upcoming events

Cloudant Service Broker Interruption
Starts April 22, 2020 11:30 PM



Catalog

- [ ] Containers
- [ ] Networking
- [ ] Storage
- [ ] AI
- [ ] Analytics
- [ ] Databases
- [ ] Developer Tools
- [ ] Integration
- [x] Internet of Things
- [ ] Security and Identity
- [ ] Web and Mobile
- [ ] Web and Application

Services

Explore our broad portfolio of managed services for infrastructure, developer tools, and more to build your apps on the public cloud.

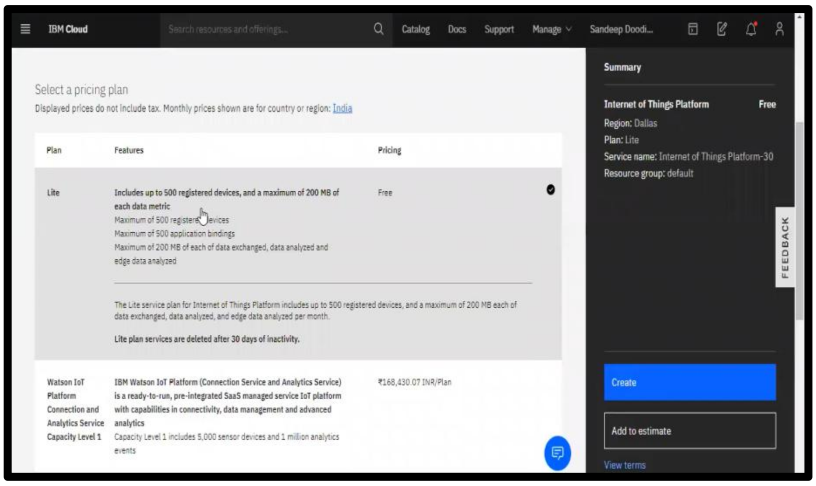
Filters: Internet of Things × Clear all

Internet of Things 8 items

Internet of Things Platform
IBM • Internet of Things
This service is the hub of all things IBM IoT, it is where you can set up and manage your connected devices so that your apps can acce...
Lite • Free • IAM-enabled

AT&T Flow Designer
Third party • Internet of Things
Design, Build and Deploy IoT Solutions in Minutes
Free

AT&T IoT Data Plans
Third party • Internet of Things
Launch your IoT product fast with IoT data plans
Free



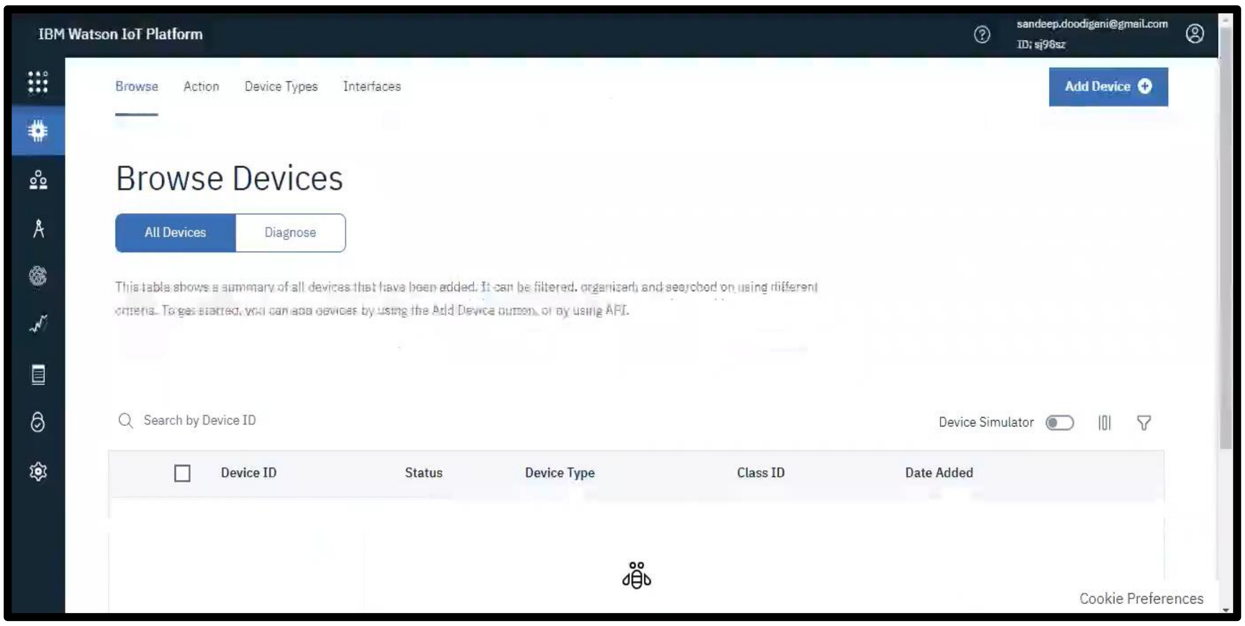
Select a pricing plan

Displayed prices do not include tax. Monthly prices shown are for country or region: India

Plan	Features	Pricing
Lite	Includes up to 500 registered devices, and a maximum of 200 MB of each data metric Maximum of 500 registered devices Maximum of 500 application bindings Maximum of 200 MB of each of data exchanged, data analyzed and edge data analyzed The Lite service plan for Internet of Things Platform includes up to 500 registered devices, and a maximum of 200 MB each of data exchanged, data analyzed, and edge data analyzed per month. **Lite plan services are deleted after 30 days of inactivity.**	Free
Watson IoT Platform Connection and Analytics Service Capacity Level 1	IBM Watson IoT Platform (Connection Service and Analytics Service) is a ready-to-run, pre-integrated SaaS managed service IoT platform with capabilities in connectivity, data management and advanced analytics Capacity Level 1 includes 5,000 sensor devices and 1 million analytics events	₹168,430.07 INR/Plan

Summary

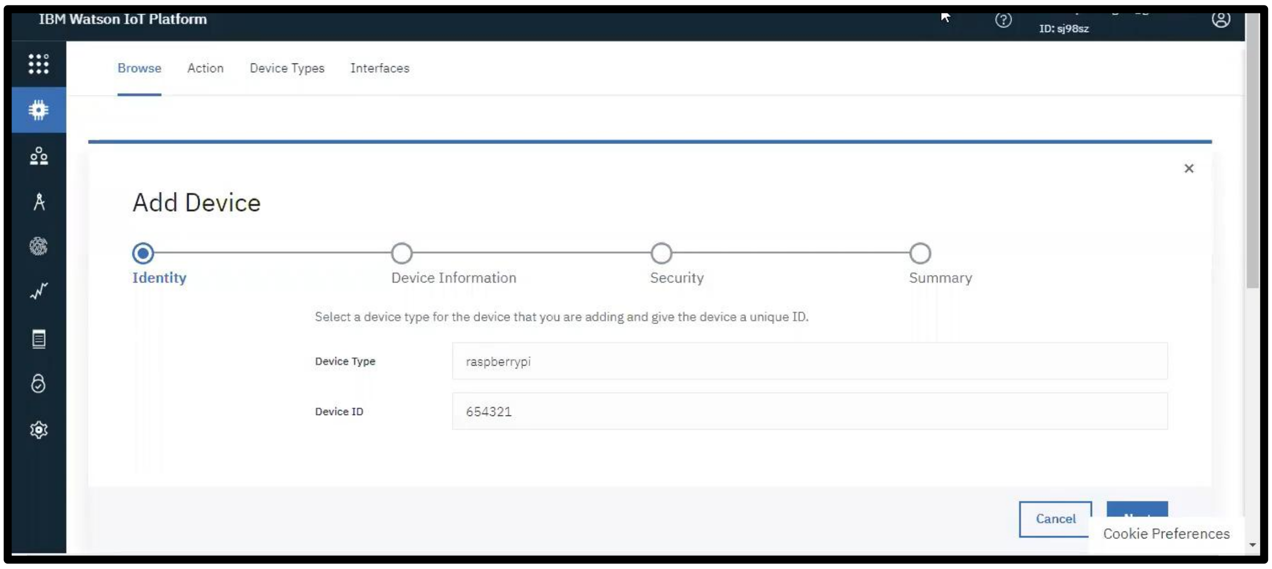
Internet of Things Platform Free
Region: Dallas
Plan: Lite
Service name: Internet of Things Platform-30
Resource group: default



Browse Devices

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Device ID	Status	Device Type	Class ID	Date Added

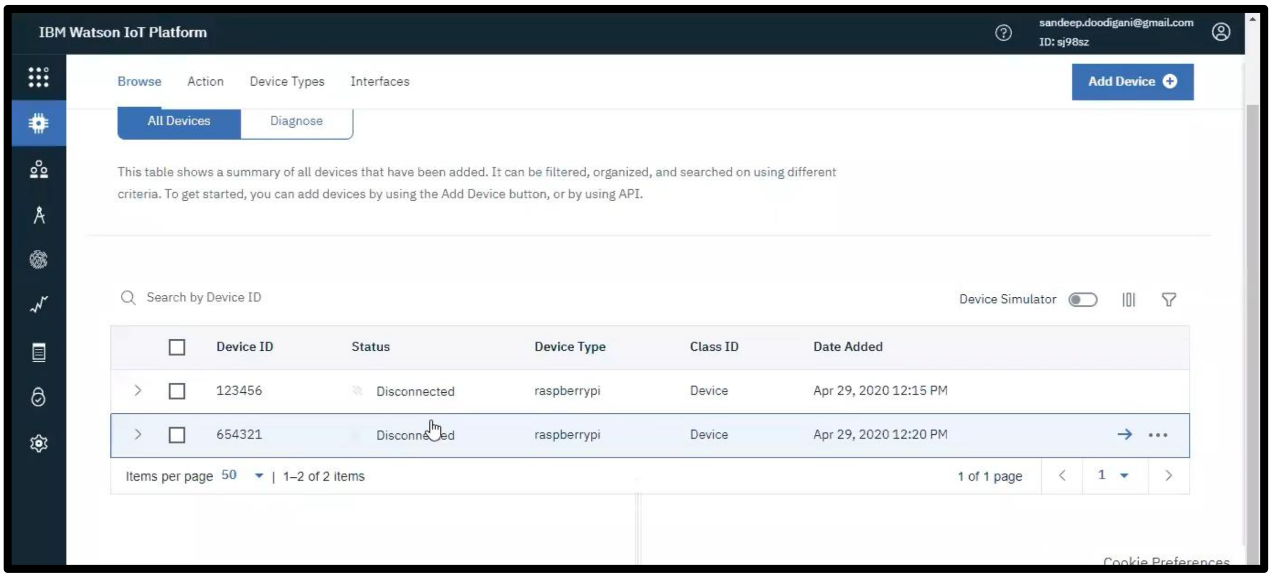


Add Device

Select a device type for the device that you are adding and give the device a unique ID.

Device Type: raspberrypi

Device ID: 654321



This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

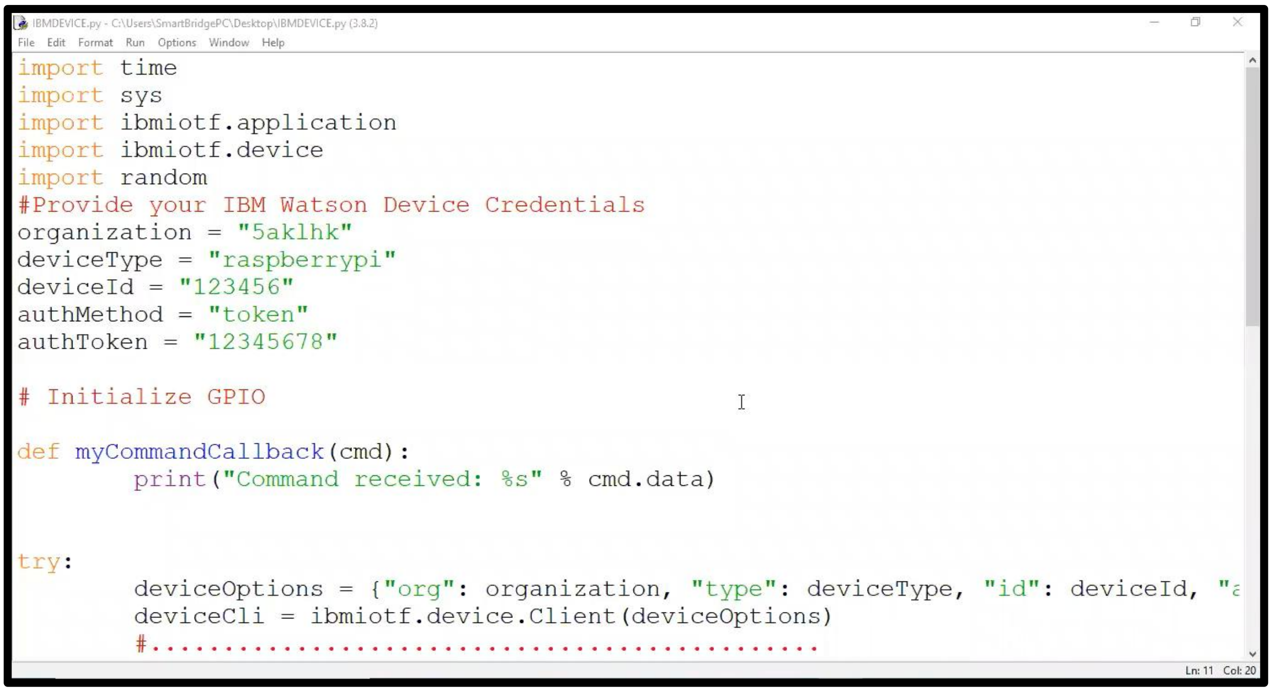
Device ID	Status	Device Type	Class ID	Date Added
123456	Disconnected	raspberrypi	Device	Apr 29, 2020 12:15 PM
654321	Disconnected	raspberrypi	Device	Apr 29, 2020 12:20 PM

Items per page 50 | 1–2 of 2 items 1 of 1 page 1

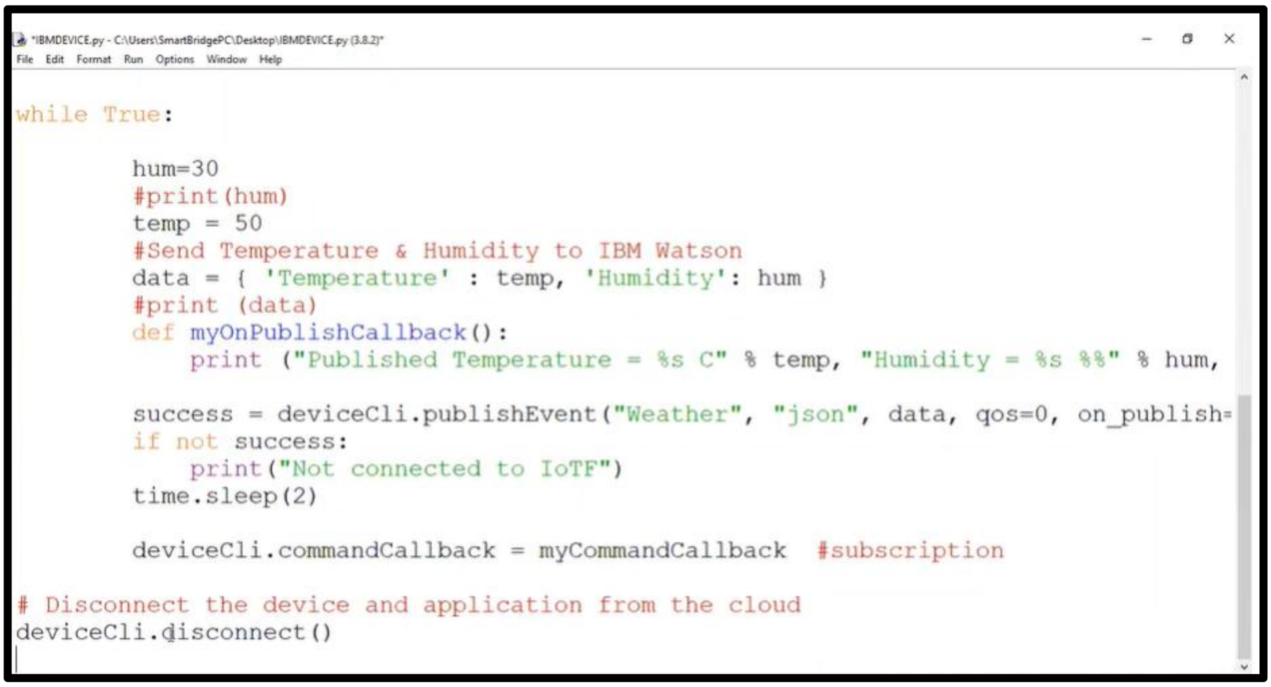
Figure 13: IBM account Creation & IoT Service

2.1.2 Connecting to IBM IoT Platform and sending sensor data using python code:

- Previously we created an IoT Platform and connected to Virtual IBM IoT simulator.
- Now we using python code to perform IoT
- By using Device data in the IoT Platform we can send data by using the Python code.
- To use Python code, we have to install some library files.
- We can check whether the required library files are available in our laptop/Computer by using command prompt.
- For installation use **pip install ibmiotf** in command prompt


IBMDEVICE.py - C:\Users\SmartBridgePC\Desktop\IBMDEVICE.py (3.8.2)
File Edit Format Run Options Window Help
Ln: 11 Col: 20

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "5aklhk"
deviceType = "raspberrypi"
deviceId = "123456"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO

def myCommandCallback(cmd):
        print("Command received: %s" % cmd.data)


try:
        deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "a
        deviceCli = ibmiotf.device.Client(deviceOptions)
        #..............................................
```


IBMDEVICE.py - C:\Users\SmartBridgePC\Desktop\IBMDEVICE.py (3.8.2)
File Edit Format Run Options Window Help

```
while True:

        hum=30
        #print(hum)
        temp = 50
        #Send Temperature & Humidity to IBM Watson
        data = { 'Temperature' : temp, 'Humidity': hum }
        #print (data)
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % hum,

        success = deviceCli.publishEvent("Weather", "json", data, qos=0, on_publish=
        if not success:
            print("Not connected to IoTF")
        time.sleep(2)

        deviceCli.commandCallback = myCommandCallback  #subscription

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

Figure 14: Python Programming to send data

CHAPTER 03

3.0 IBM Cloud Services

3.0.0 IBM Cloudant:

What is IBM Cloudant?
IBM Cloudant is a fully managed JSON document database that offers independent serverless scaling of throughput capacity and storage.

How to open IBM Cloudant and use it?
First fall open IBM cloud and there will open a dashboard page and click on catalog and there will see all the services of IBM. and click on catalog then left side we see services option click on it and search for Cloudant, and there will see a Cloudant database and click on it.

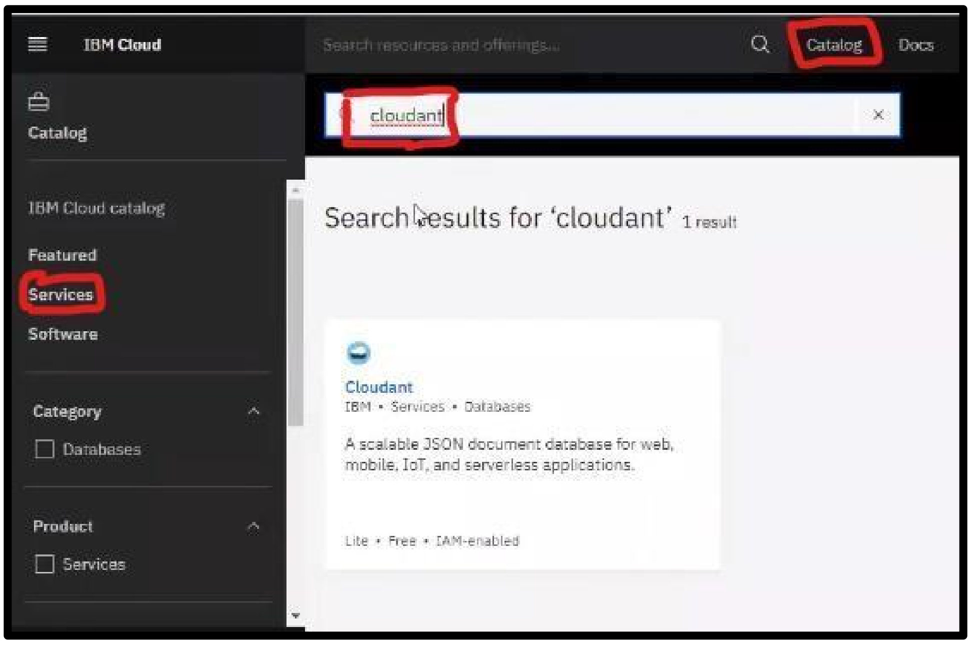
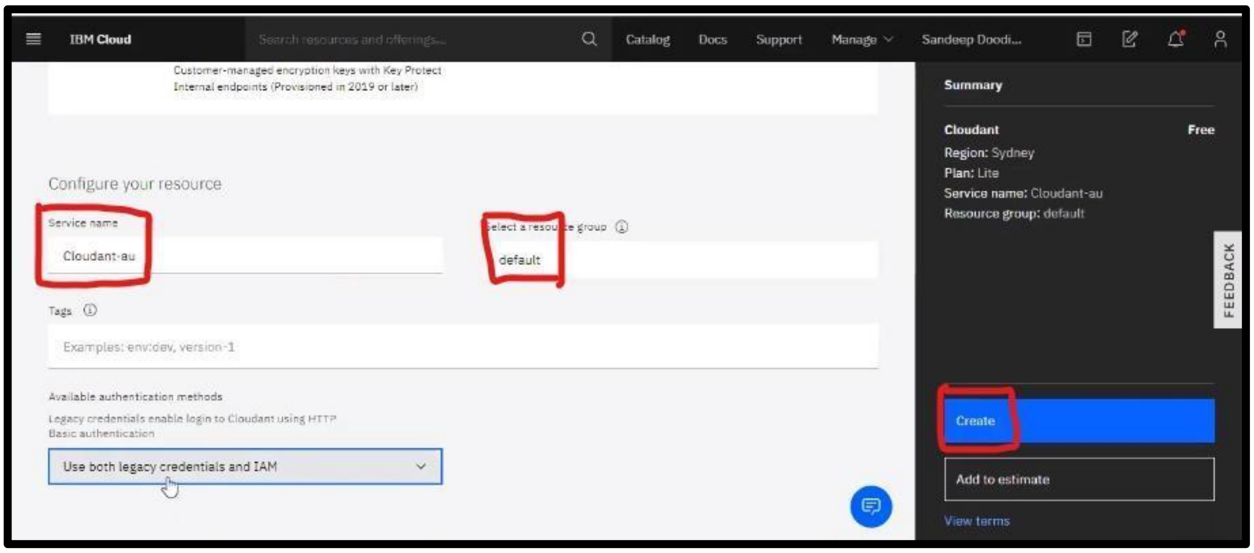

IBM Cloud
Search resources and offerings...
Catalog
Docs
Catalog
cloudant
IBM Cloud catalog
Featured
Services
Software
Category
Databases
Product
Services
Search results for 'cloudant' 1 result
Cloudant
IBM • Services • Databases
A scalable JSON document database for web, mobile, IoT, and serverless applications.
Lite • Free • IAM-enabled

Fig18: Cloudant in catalog page

In Cloudant we store the data in JSON format. The Cloudant page will be open and fill necessary credentials and click on create.



Customer-managed encryption keys with Key Protect
Internal endpoints (Provisioned in 2019 or later)

Configure your resource

Service name: Cloudant-au

Select a resource group: default

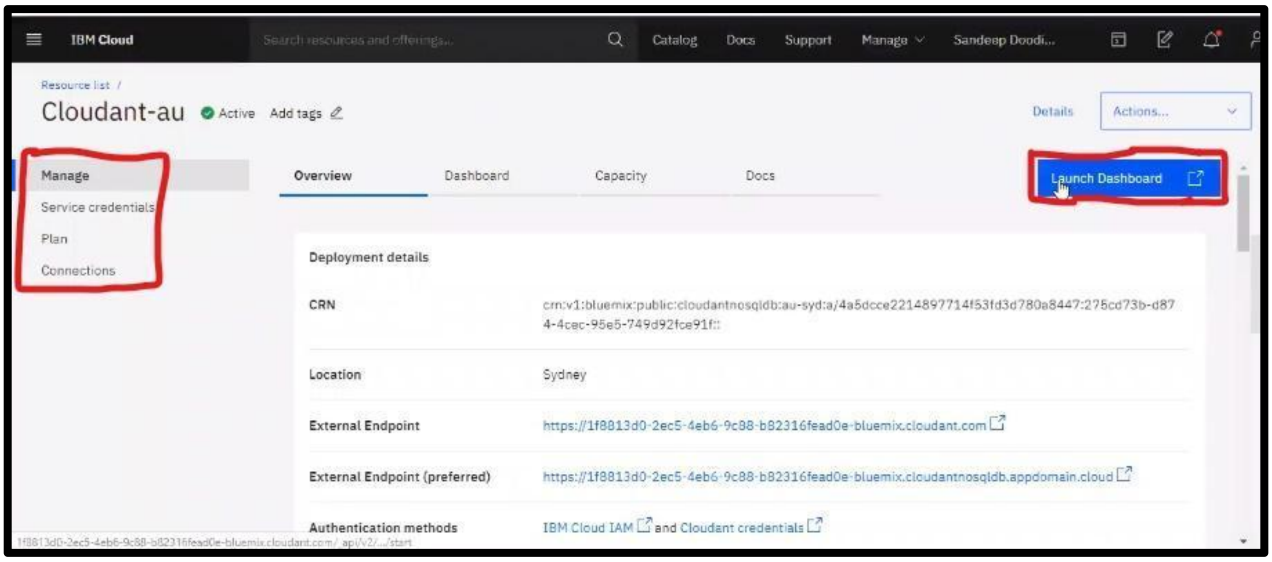
Tags: Examples: env:dev, version-1

Available authentication methods
Legacy credentials enable login to Cloudant using HTTP Basic authentication
Use both legacy credentials and IAM

Summary
Cloudant Free
Region: Sydney
Plan: Lite
Service name: Cloudant-au
Resource group: default

Fig 19: Cloudant database creating page

Now open Cloudant service and click on the launch dashboard page.



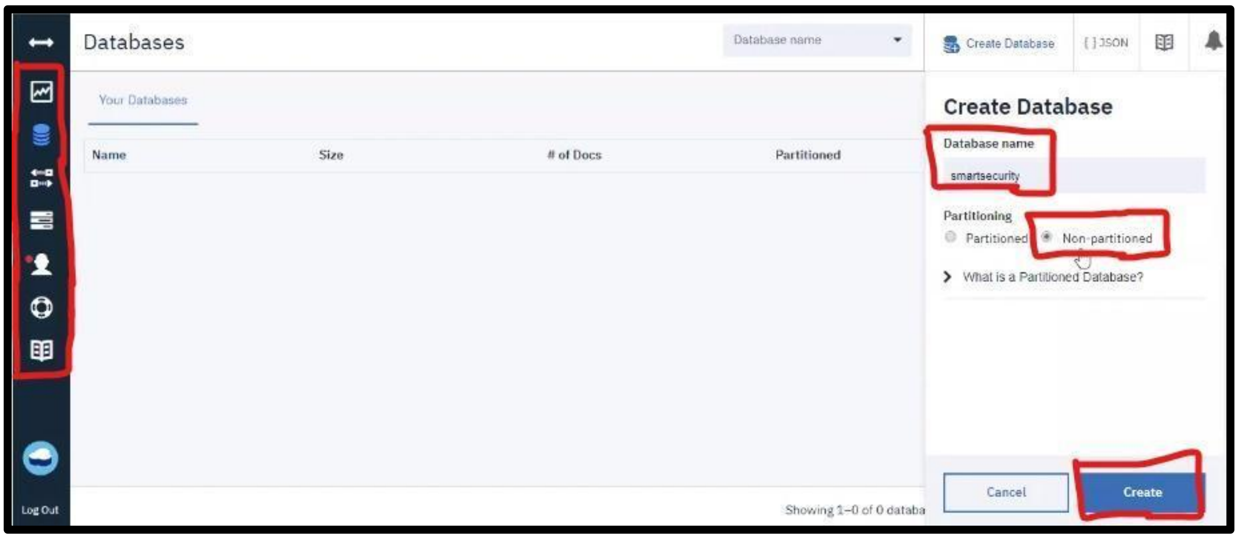
Cloudant-au Active Add tags

Deployment details

CRN	crn:v1:bluemix:public:cloudantnosqldb:au-syd:a/4a5dcce2214897714f53fd3d780a8447:275cd73b-d874-4cec-95e5-749d92fce91f::
Location	Sydney
External Endpoint	https://1f8813d0-2ec5-4eb6-9c88-b82316fead0e-bluemix.cloudant.com
External Endpoint (preferred)	https://1f8813d0-2ec5-4eb6-9c88-b82316fead0e-bluemix.cloudantnosqldb.appdomain.cloud
Authentication methods	IBM Cloud IAM and Cloudant credentials

Fig 20: Cloudant launch dashboard

And the database page will be open for us. and click on create database and there a pop page will be open on right side and create a database name and there we have an option is partitioning in that we have to click on non-partitioned and then click on create button.



Name	Size	# of Docs	Partitioned

Create Database

Database name: smartsecurity

Partitioning: Partitioned [✓ Non-partitioned]

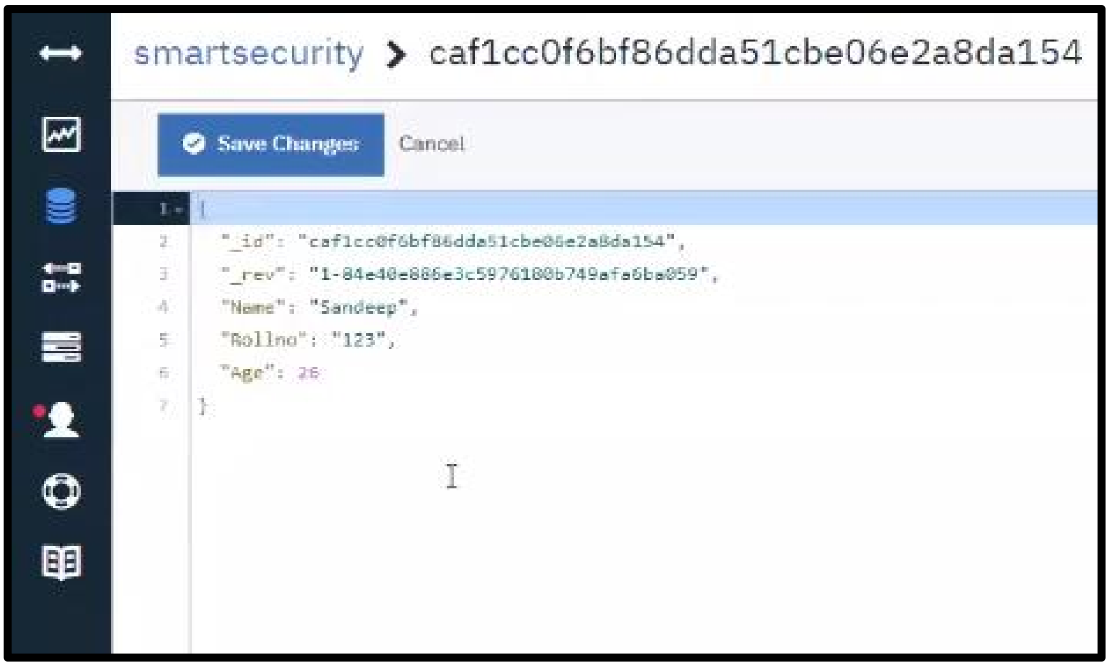
What is a Partitioned Database?

Cancel Create

Showing 1–0 of 0 databa

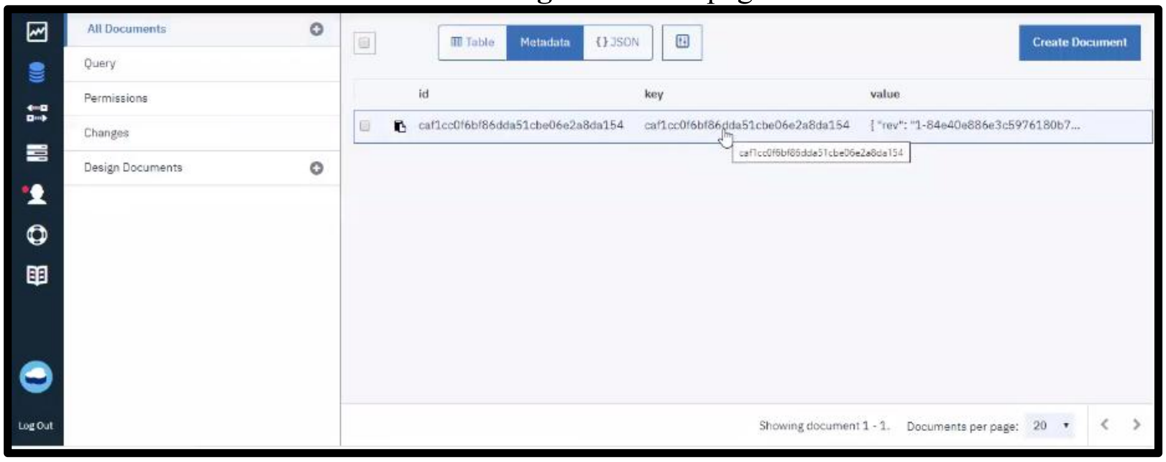
Fig 21: creating database

And we create our data here and documents will be created in it. To run the python code in it. We have to open database page and check if data is created or not.



```
{
  "_id": "caf1cc0f6bf86dda51cbe06e2a8da154",
  "_rev": "1-84e40e886e3c5976180b749afa6ba059",
  "Name": "Sandeep",
  "Rollno": "123",
  "Age": 26
}
```

Fig 22: Editor page



id	key	value
caf1cc0f6bf86dda51cbe06e2a8da154	caf1cc0f6bf86dda51cbe06e2a8da154	{ "rev": "1-84e40e886e3c5976180b7...

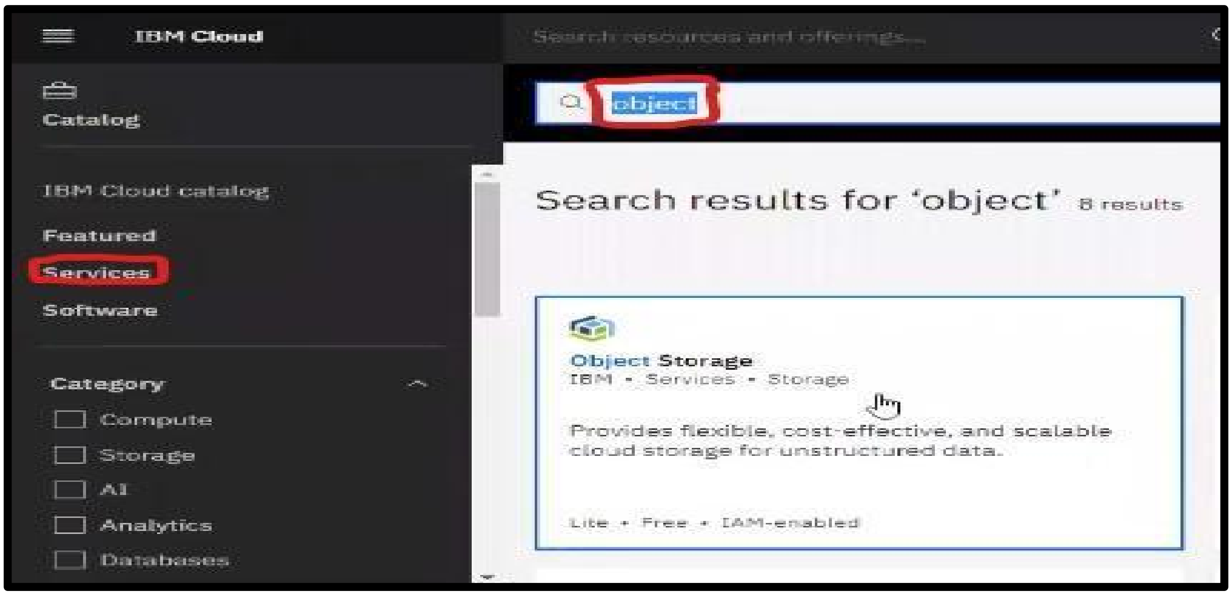
Showing document 1 - 1. Documents per page: 20

Fig 23: Data is created in json format

3.0.1 IBM Object Storage:

What is IBM object storage?
IBM Object Storage makes it possible to store practically limitless amounts of data, simply and cost effectively. It is commonly used for data archiving and backup; for web and mobile applications; and as scalable, persistent storage for analytics.

How to open IBM Object Storage and use it?
First fall open IBM cloud and there will open a dashboard page and click on catalog and there will see all the services of IBM. and click on catalog then left side we see a services option click on it. Search for Object Storage, there we observe Object Storage then click on it.



Search results for 'object' 8 results

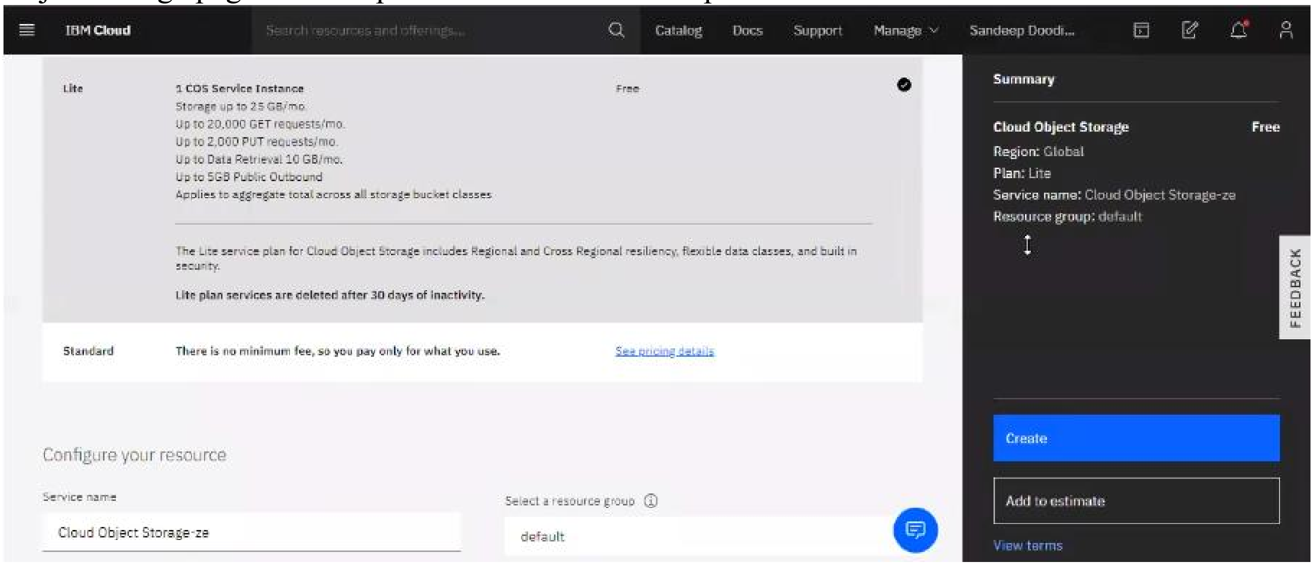
Object Storage
IBM • Services • Storage

Provides flexible, cost-effective, and scalable cloud storage for unstructured data.

Lite • Free • IAM-enabled

Fig 24: Object storage

Object storage page will be open then click on create option then service will be created.



Lite	**1 COS Service Instance** Storage up to 25 GB/mo. Up to 20,000 GET requests/mo. Up to 2,000 PUT requests/mo. Up to Data Retrieval 10 GB/mo. Up to 5GB Public Outbound Applies to aggregate total across all storage bucket classes The Lite service plan for Cloud Object Storage includes Regional and Cross Regional resiliency, flexible data classes, and built in security. **Lite plan services are deleted after 30 days of inactivity.**	Free
Standard	There is no minimum fee, so you pay only for what you use.	See pricing details

Configure your resource

Service name: Cloud Object Storage-ze

Select a resource group: default

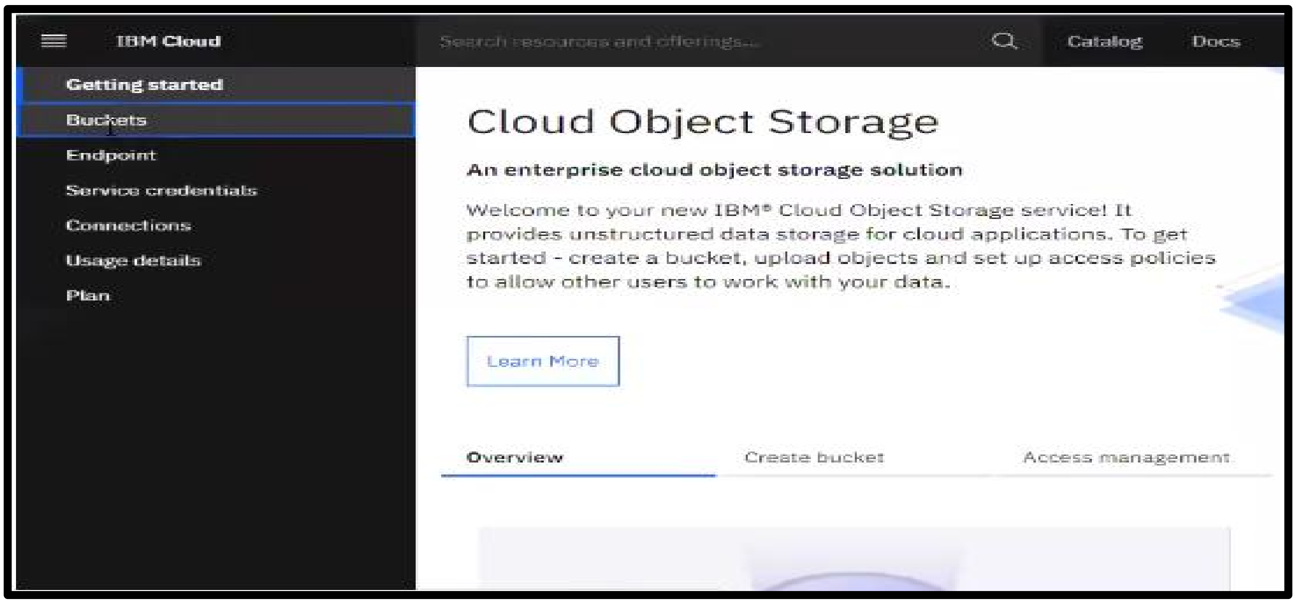
Summary

Cloud Object Storage Free
Region: Global
Plan: Lite
Service name: Cloud Object Storage-ze
Resource group: default

Create
Add to estimate
View terms

Fig 25: Creating object storage

Object storage will be open and click on buckets and then click on create bucket button there are different kinds of buckets. We have to click on a custom bucket and the bucket should always be a unique name and click on the create bucket.



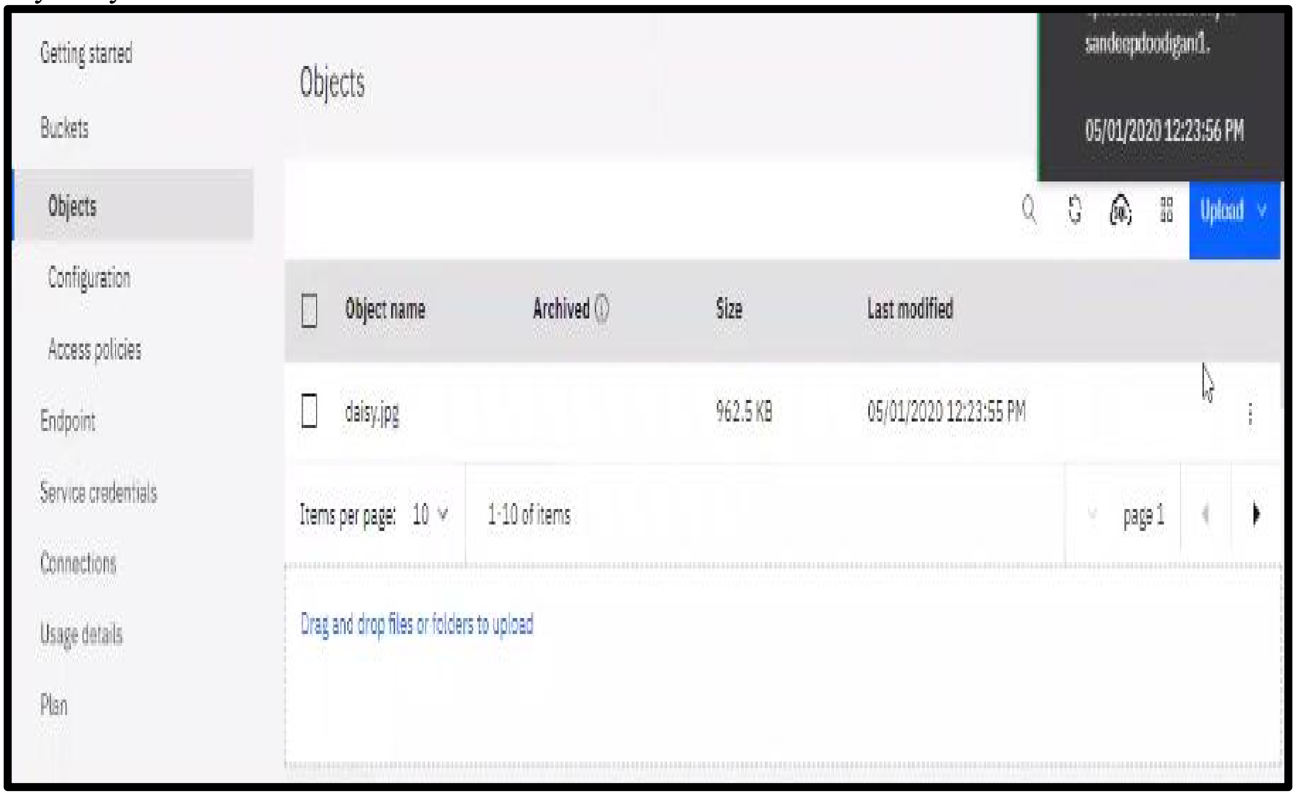
Cloud Object Storage

An enterprise cloud object storage solution

Welcome to your new IBM® Cloud Object Storage service! It provides unstructured data storage for cloud applications. To get started - create a bucket, upload objects and set up access policies to allow other users to work with your data.

Fig 26: Creating buckets

There will be a drag and drop option then click on it. A pop page will be opened and select any file you want to store in it.



Object name	Archived	Size	Last modified
daisy.jpg		962.5 KB	05/01/2020 12:23:55 PM

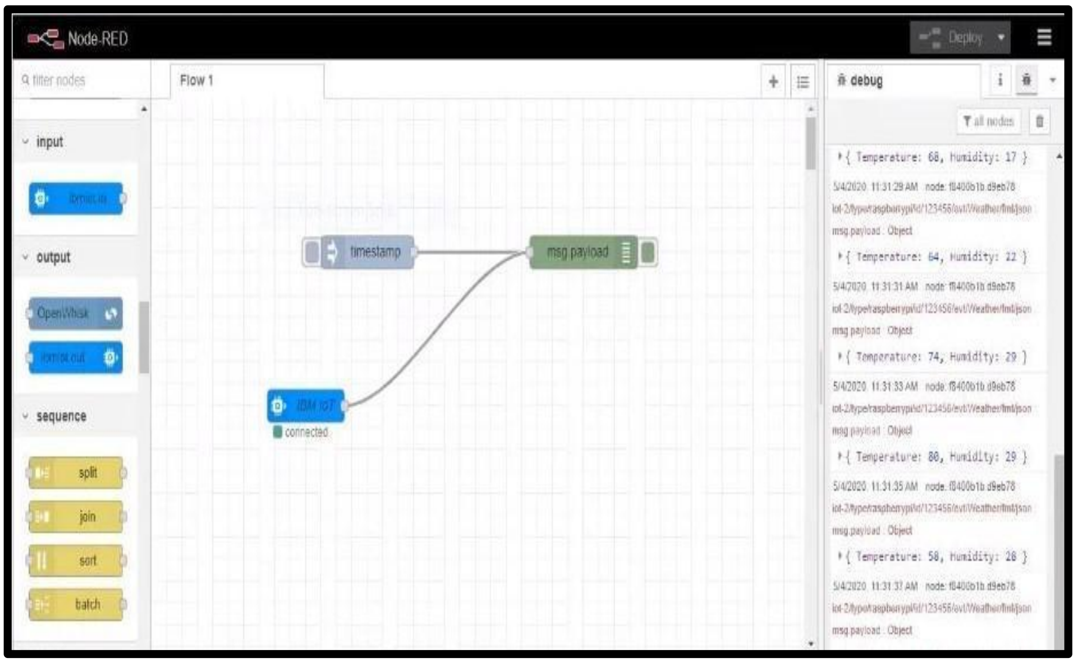
Fig 27: Data is stored

3.1 NODE RED

WHAT IS Node Red?
Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways. It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click.

3.1.0 Retrieving Data from IBM IoT Platform:

First fall open IBM cloud and there will open a dashboard page and click on catalog and there will see all the services of IBM. and click on catalog then left side we see software option click on it. and search for Node-red and there will see a Node-Red App then click on it. Node-Red page will be open and the back end operates under Cloudant database. Node-Red has been developed on Node. JS and click on create and give some necessary credentials in it. Then click on Next and duplicate the same page and come back to the dashboard page then click on services that click on the internet of things platform then click on the launch button. So, it will launch the IBM IoT platform. In the node-red page click on IBM cloud and click on cloud foundry apps in that click on Node-Red and then click on Visit App URL.then welcome page will be open of Node-Red and click on next finally click on finish. And click on Go to Node-Red flow Editor then open the Node-Red tool page. On the Node-Red left side we have filter nodes and the middle page we have a work bench and on the right side we have console and every time we have to click on Deploy to save the flow. Connect necessary nodes and click on Deploy to save the flow of the Node-Red, in right side of the corner we have two buttons one is Node information and other one is Debug messages and then click on the button on work bench then we see the debug information on the right side of the debug message area. We have to Retrieve Data from IBM IoT Platform and we have installed some libraries in it. Then click on the menu button and go to manage palette and click on install and search for IBM IoT and click on install and then click on again install. On the left side we have seen IBM IoT nodes and drag the node to the work bench and double tap on the IBM IoT and insert API KEY and API token in it and click on add and come back to the same page then add some credentials in it. and then connected to the IBM IoT platform and whatever data coming in IoT platform that data comes to node-red flow also.

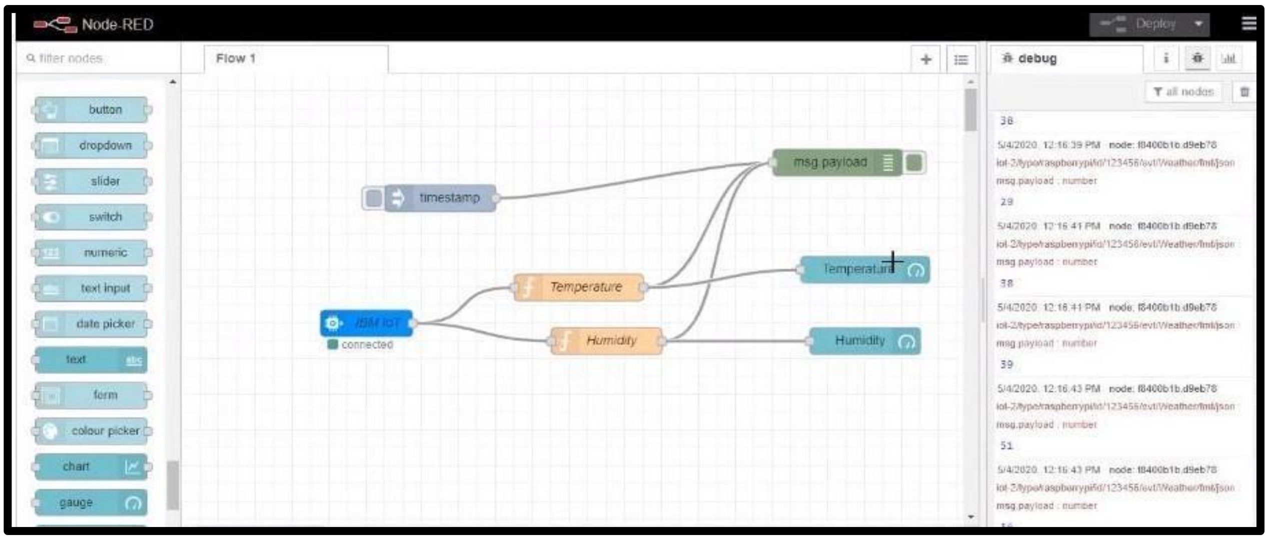


```
{ Temperature: 68, Humidity: 17 }
5/4/2020 11:31:29 AM  node: f8400b1b.d9eb78
iot-2/type/raspberrypi/id/123456/evt/Weather/fmt/json
msg.payload : Object
{ Temperature: 64, Humidity: 22 }
5/4/2020 11:31:31 AM  node: f8400b1b.d9eb78
iot-2/type/raspberrypi/id/123456/evt/Weather/fmt/json
msg.payload : Object
{ Temperature: 74, Humidity: 29 }
5/4/2020 11:31:33 AM  node: f8400b1b.d9eb78
iot-2/type/raspberrypi/id/123456/evt/Weather/fmt/json
msg.payload : Object
{ Temperature: 80, Humidity: 29 }
5/4/2020 11:31:35 AM  node: f8400b1b.d9eb78
iot-2/type/raspberrypi/id/123456/evt/Weather/fmt/json
msg.payload : Object
{ Temperature: 58, Humidity: 28 }
5/4/2020 11:31:37 AM  node: f8400b1b.d9eb78
iot-2/type/raspberrypi/id/123456/evt/Weather/fmt/json
msg.payload : Object
```

Fig 28: Node-Red flow diagram

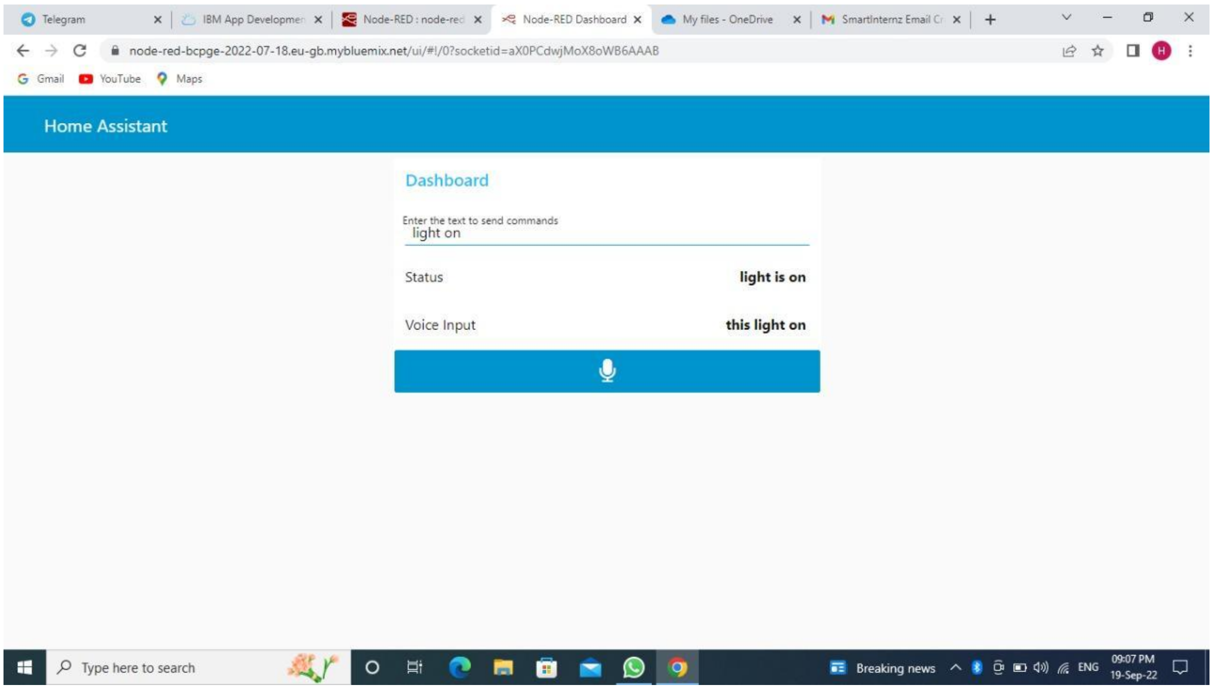
3.1.1 Sending Command to the Device:

Sending commands to connected devices by using an IoT simulator. By adding additional functions to it, temperature and humidity functions to the message. Payload sees the same values in the IoT simulator. Whatever information comes in the node-red. If we want to see the interface of UI then copy the node-red URL and open the new tab and paste URL in Web and add additional term UI in the web lastly and then press enter it is shown as figure.



```
38
5/4/2020 12:16:39 PM  node: f8400b1b.d9eb78
iot-2/type/raspberrypi/id/123456/evt/Weather/fmt/json
msg.payload : number
29
5/4/2020 12:16:41 PM  node: f8400b1b.d9eb78
iot-2/type/raspberrypi/id/123456/evt/Weather/fmt/json
msg.payload : number
38
5/4/2020 12:16:41 PM  node: f8400b1b.d9eb78
iot-2/type/raspberrypi/id/123456/evt/Weather/fmt/json
msg.payload : number
39
5/4/2020 12:16:43 PM  node: f8400b1b.d9eb78
iot-2/type/raspberrypi/id/123456/evt/Weather/fmt/json
msg.payload : number
51
5/4/2020 12:16:43 PM  node: f8400b1b.d9eb78
iot-2/type/raspberrypi/id/123456/evt/Weather/fmt/json
msg.payload : number
```

Fig 30: Sending command to device


Home Assistant

Dashboard

Enter the text to send commands
light on

Status light is on

Voice Input this light on

Fig 31: Receiving data from web

3.2 IBM Text To Speech and Speech To Text

What is IBM text to speech and speech to text?
IBM Watson Speech to Text (STT) is a service on the IBM Cloud that enables you to easily convert audio and voice into written text and vice versa for speech to text.

How to open IBM text to speech and use it?
First fall open IBM cloud and there will open a dashboard page and click on catalog and there will see all the services of IBM. and click on catalog then left side we see services option, in category select AI and there we see text to speech service then click on it. The service will be open and click on create. Then the text to speech page will be open on that page on the left side. We have service credentials that copy those credentials one side. and write the necessary python code in the python shell and place those credentials in that code and run the code successfully .in desktop we see a text to speech file on it.

How to open IBM speech to text and use it?
First fall open IBM cloud and there will open a dashboard page and click on catalog and there will see all the services of IBM and click on catalog then left side we see services optioning category select AI and there we see speech to text service then click on it. These services will be open and click on create. Then the speech to text page will be open on that page on the left side we have service credentials copy that credentials one side. Write a necessary python code in the python shell and place those credentials in that code and run the code successfully. In the python output shell we see the speech as text format.

3.3 Text To Speech and Speech To Text Using Node Red

I want to compare the text (generated from speech to text) that I get from the microphone with the data in the database.

How to open text to speech and speech to text using Node Red and use it?

First fall open IBM cloud and there will open a dashboard page there we will see cloud foundry apps in that we have Node-Red service then click on it then select visit url and navigate to the Node-Red page and select go to your Node-Red flow editor and duplicate the same page and again go to back to dashboard page and select services in that we have text to speech click on it and page will be open there you will see API keys and URL copy that it one side and in that Node-Red page select text to speech function then double click on it and enter the API keys and URL in it then click on done. Then again double click on it there we see some additional options in it example like language selecting and voice of language options then which language you want and we have to enter input data in it add inject node in it and double click on it and select the payload as string and give some text in that and click on done and connect both of them. We need to have some play audio nodes then select and drag to the work bench and connect to text to speech node and then we have to save the nodes then click on deploy. so if we want to give text to the speech node then click on the input node, so we can hear the voice as output. In this way we can convert text as speech and vice versa for the speech to text in that we are using a microphone in it for recording, recording is done then leaving the microphone button and it passes through the speech to text, whatever you speak that speech comes as text in debug section.

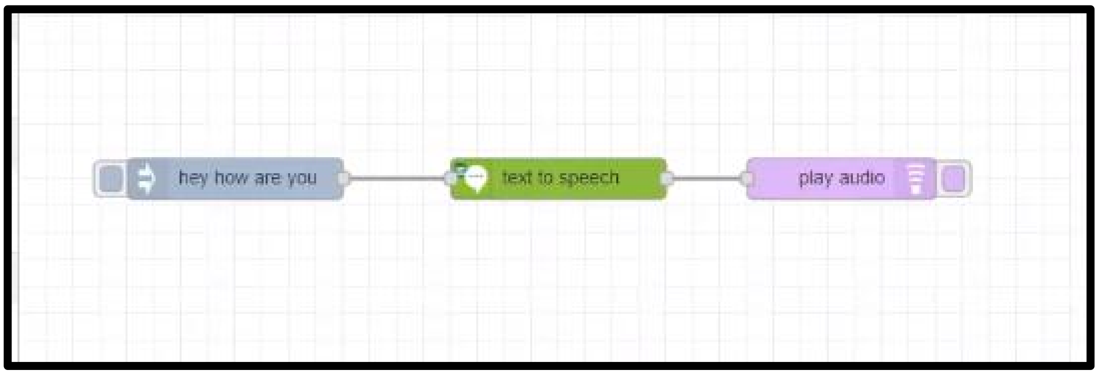
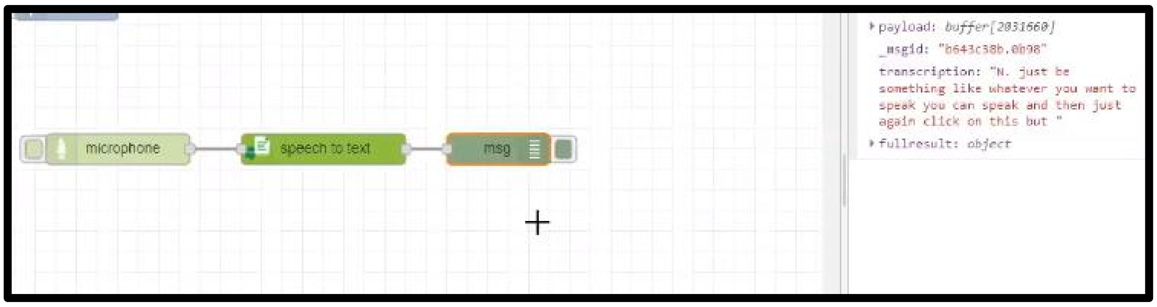


Fig 33: Text to speech



```
▸ payload: buffer[2031660]
  _msgid: "b643c38b.0b98"
  transcription: "N. just be
  something like whatever you want to
  speak you can speak and then just
  again click on this but "
▸ fullresult: object
```

Fig 34: Speech to text

3.4 Watson Assistant using python code

What is Watson assistant?
Watson Assistant is a conversation AI platform that helps you provide customers with fast, straightforward and accurate answers to their questions, across any application, device or channel. ... Watson Assistant is more than a Chabot.

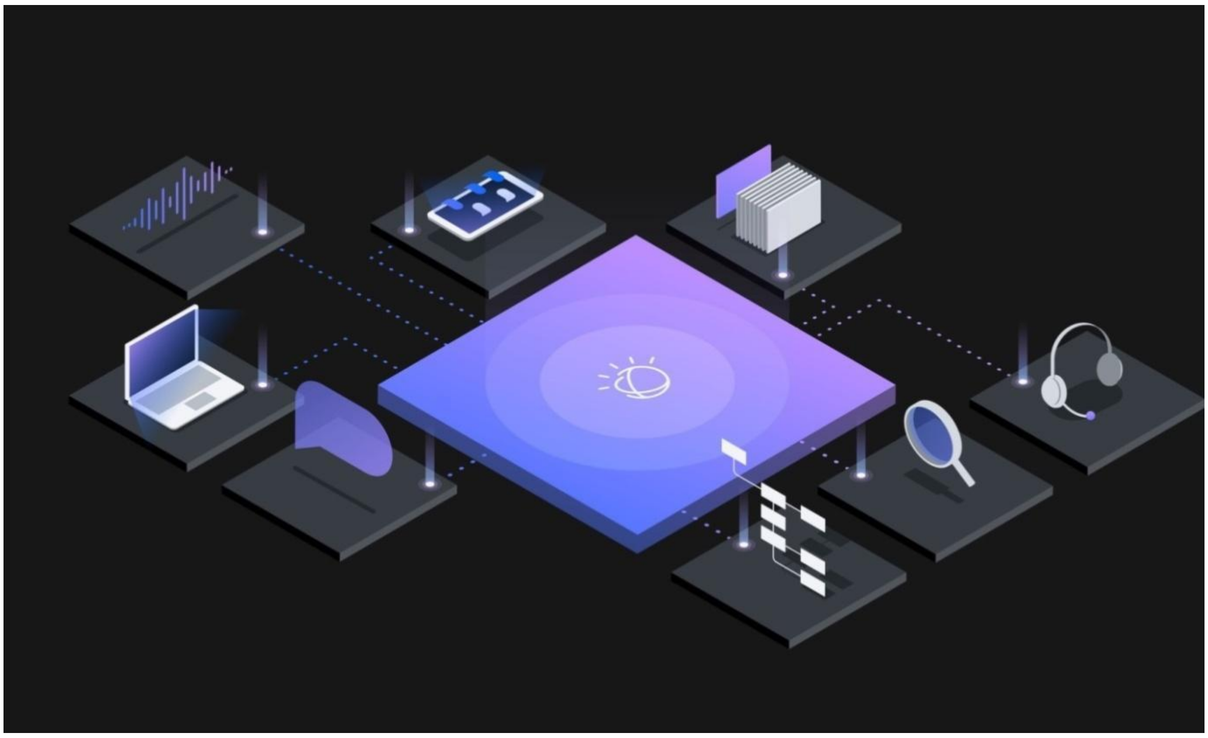
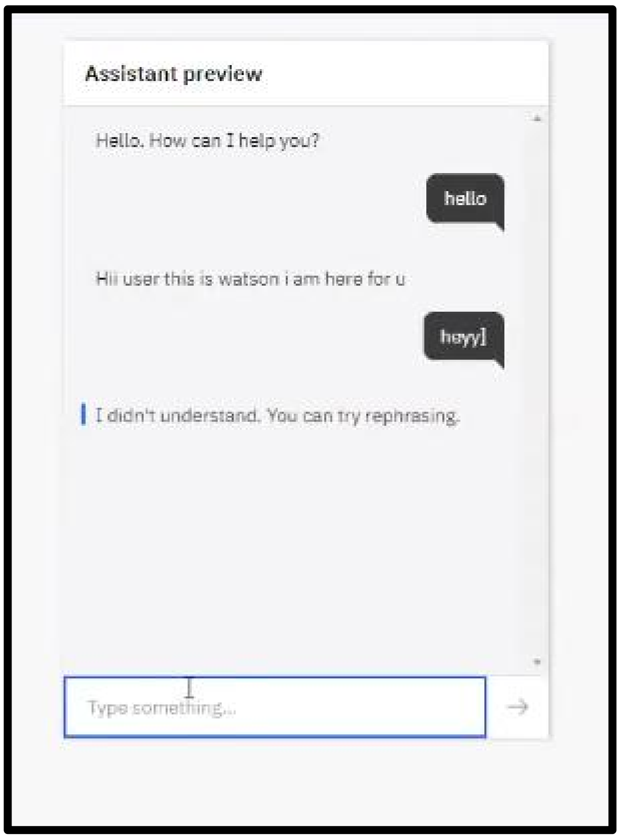


Fig 35: Watson assistant AI platform

How to open Watson Assistant and work it?

First fall open IBM cloud and there will open a dashboard page there we will see a services button in service button we have Watson assistant click on it Watson assistant page will be open there we have some credentials in it copy them in one side and paste this credentials in the python code when this credentials needed and click on launch Watson assistant. There you see the assistant page. We want some libraries in it and download the libraries. We have to create a session to use Watson assistant every time and copy Watson assistant code and paste the code in python shell and paste the API keys and Watson assistant keys in it and run the code in python shell then we get the session id copy that id in a notepad carefully. In API docs left side we have methods in methods we have messages in messages select send user input to assistant there we have a python code copy that code and paste the code in python shell and give necessary credentials to it, lastly, we have copied session id and place them in that code and run that python code successfully. We provide some skills to our Watson assistant then Watson assistant is trained successfully.


Assistant preview

Hello. How can I help you?

hello

Hii user this is watson i am here for u

heyy]

I didn't understand. You can try rephrasing.

Fig 36: Watson assistant preview

CHAPTER 04

4.0 Applications

4.0.0 Smart Security Using Node-RedService

Login to the IBM Cloud and open the Cloudant database service, Node-Red and object storage. Give the service credentials of the object storage. From the end point give the resiliency as regional, location as jp-tok, copy and paste the public URL. Give the service credentials to the Cloudant service to the python code. Give the database name =sample. Open the Fast2sms website and login. click on Dev API and then open read API documentation. Click on Bulk SMS API and click Get Method. Copy the URL and paste it new tab and give the API key from Fast2sms and give the mobile number.

4.1 IBM Cloud Functions

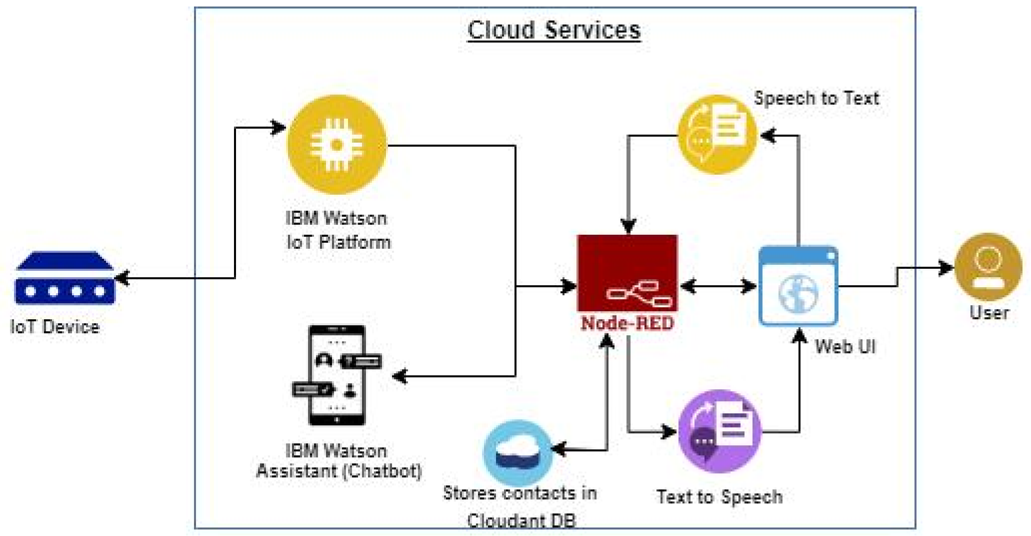
IBM Cloud Functions is a distributed compute service that executes application logic in response to requests from web or mobile apps. You can set up specific actions to occur based on HTTP-based API requests from web apps or mobile apps, and from event-based requests from services like Cloudant. Functions can run your code snippets on demand or automatically in response to events. All APIs are protected with HTTP Basic authentication. You can use the wskadmin tool to generate a new namespace and authentication. The Basic authentication credentials are in the AUTH property in your ~/.wskprops file, delimited by a colon. You can also retrieve these credentials by using the CLI running ibmcloud wsk property get --auth. For the {namespace} in the URL, the underscore character (_) can be used to specify the user's default namespace.

CHAPTER 05

5.1 Purpose of the Project

"Home automation" refers to the automatic and electronic control of household features, activity, and appliances. In simple terms, it means you can easily control the utilities and features of your home via the Internet to make life more convenient and secure, and even spend less on household bills. The main objective of home automation is to help handicapped and old aged people which will enable them to control home appliances and alert them in critical situations.

5.2 Block Diagram



5.3 Hardware/Software Designing

- Python
- IOT Open Hardware Platforms
- IOT Application Development
- IOT Cloud Platform
- IOT Communication Technologies
- IOT Communication Protocols

5.4 Advantages & Disadvantages

Advantages:

- ✓ Devices can be controlled from long distances
- ✓ Full control over all smart appliances with only one device
- ✓ Support for the older generation and disabled persons

Disadvantages:

- ✓ Reliable internet connection is crucial
- ✓ Not suitable for all type of houses
- ✓ Maintenance and repair issues

5.5 Python Code

```
#IBM Watson IOT Platform

#pip install wiotp-sdk

import wiotp.sdk.device

 import wiotp.sdk.device

import time

import os

import datetime

import random

myConfig = {

"identity":{
    "orgId": "fbeobe",
    "typeId": "NodeMCU",
            "deviceId":"12345"
},
        "auth": {
            "token":"12345678"
}
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s " % cmd.data['command'])
    m=cmd.data['command']
    if(m=="lighton"):
      print("Light is switched ON")
    elif(m=="lightoff"):
      print("Light is switched OFF")
    if(m=="fanon"):
      print("Fan is switched ON")
    elif(m=="fanoff"):
      print("Fan is switched OFF")
    print(" ")
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
temp=random.randint(-20,125)
hum=random.randint(0,100)
myData={'temperature':temp, 'humidity':hum}
```

client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)

print("Published data successfully: %s", myData)

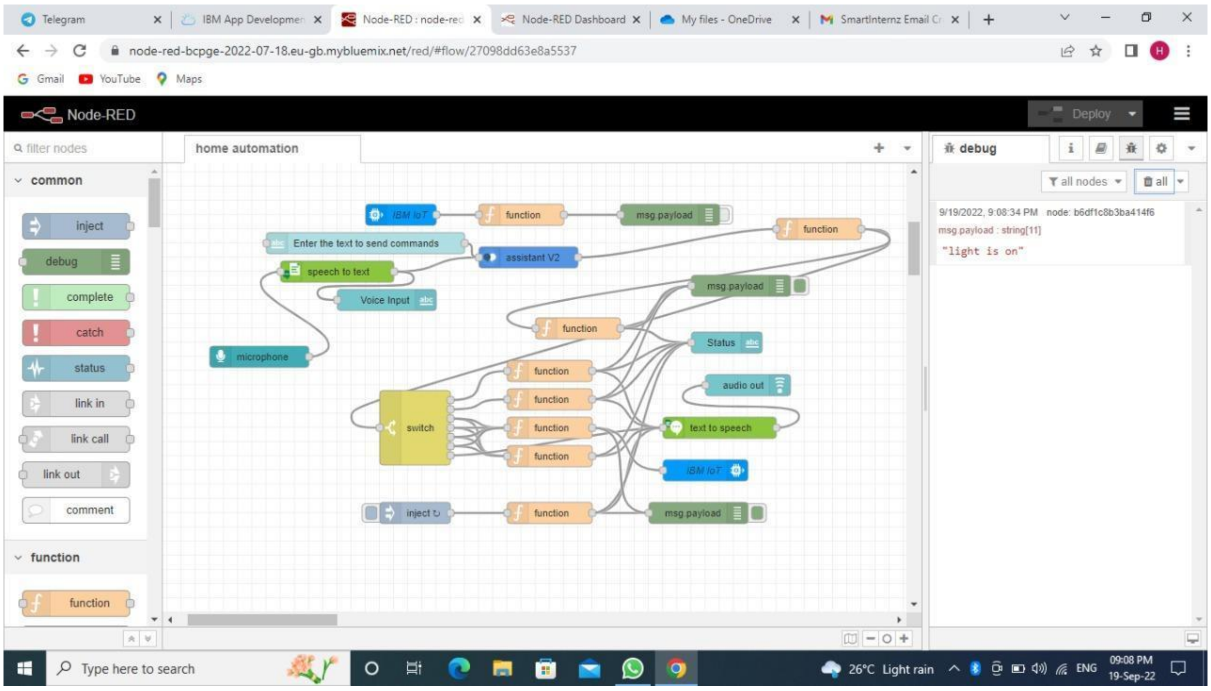
client.commandCallback = myCommandCallback

time.sleep(2)

client.disconnect()

5.5 Node-Red flow & Results

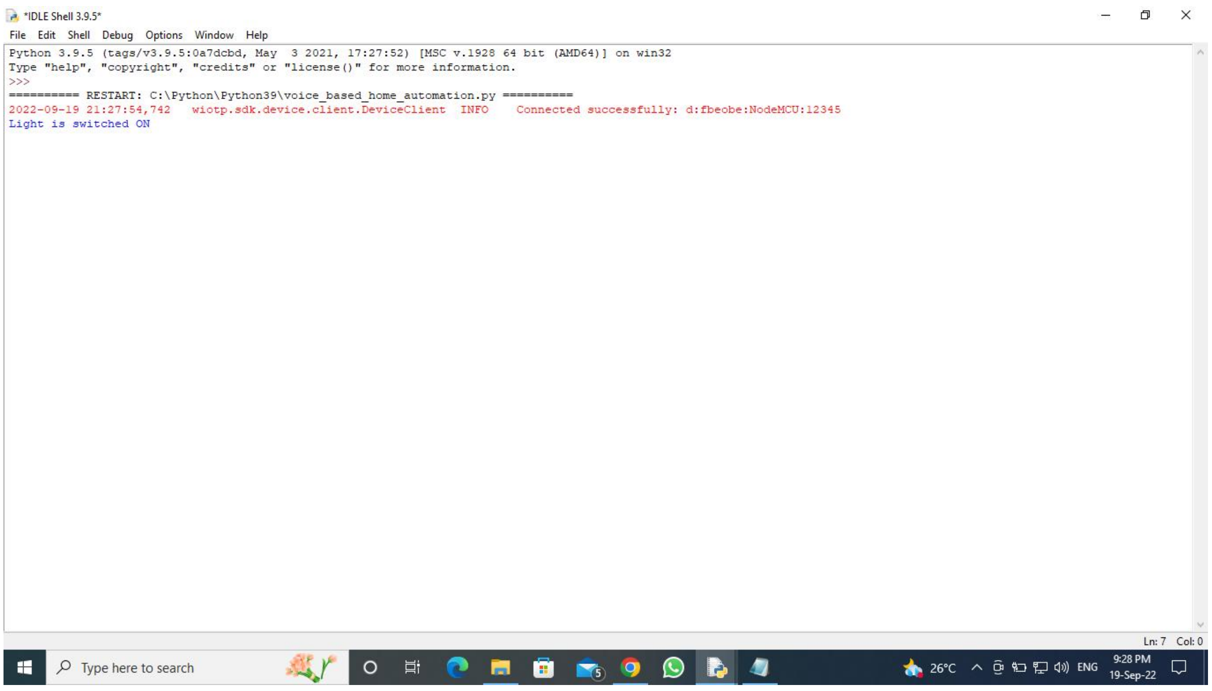
NodeRed:



```
9/19/2022, 9:08:34 PM   node: b6df1c8b3ba414f6
msg.payload : string[11]
"light is on"
```

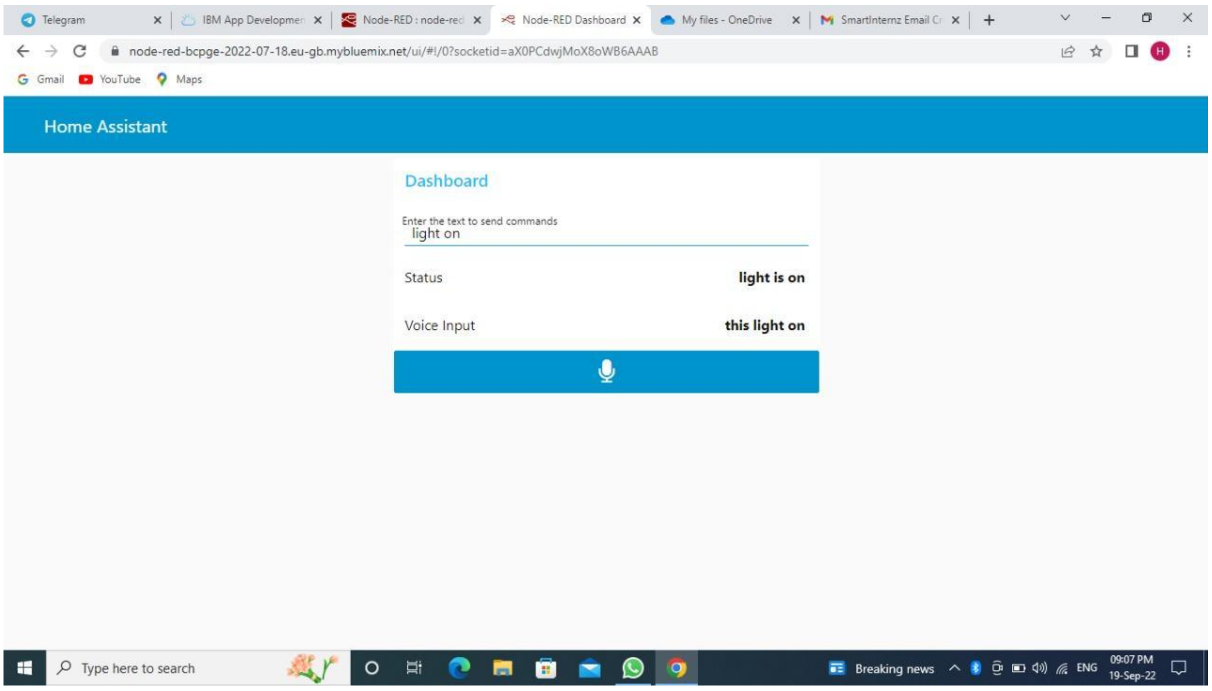
Fig: Node-Red Flow

Results



```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May  3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Python\Python39\voice_based_home_automation.py ==========
2022-09-19 21:27:54,742   wiotp.sdk.device.client.DeviceClient  INFO    Connected successfully: d:fbeobe:NodeMCU:12345
Light is switched ON
```

Node-RedUI:


node-red-bcpge-2022-07-18.eu-gb.mybluemix.net/ui/#!/0?socketid=aX0PCdwjMoX8oWB6AAAB
Home Assistant
Dashboard

Enter the text to send commands
light on

Status light is on

Voice Input this light on

fig: Node-Red flow output

CHAPTER 06

6.0 Conclusion

In conclusion, I am well satisfied with my training. I have learned many new concepts, acquired a number of new technical skills and improved another group of existing skills. What I liked most about my training is that it is very strongly related to new emerging technology. This refutes the common saying that very little of the materials taught in university engineering courses is used by engineers working in the labor market. This dependency (relationship) is clearest in engineering design. I may count the technical skills that I learned or improved at the training site, other than those gained at college. At last, I hereby conclude that I have successfully completed my industrial training in Smart Bridge and gained knowledge.