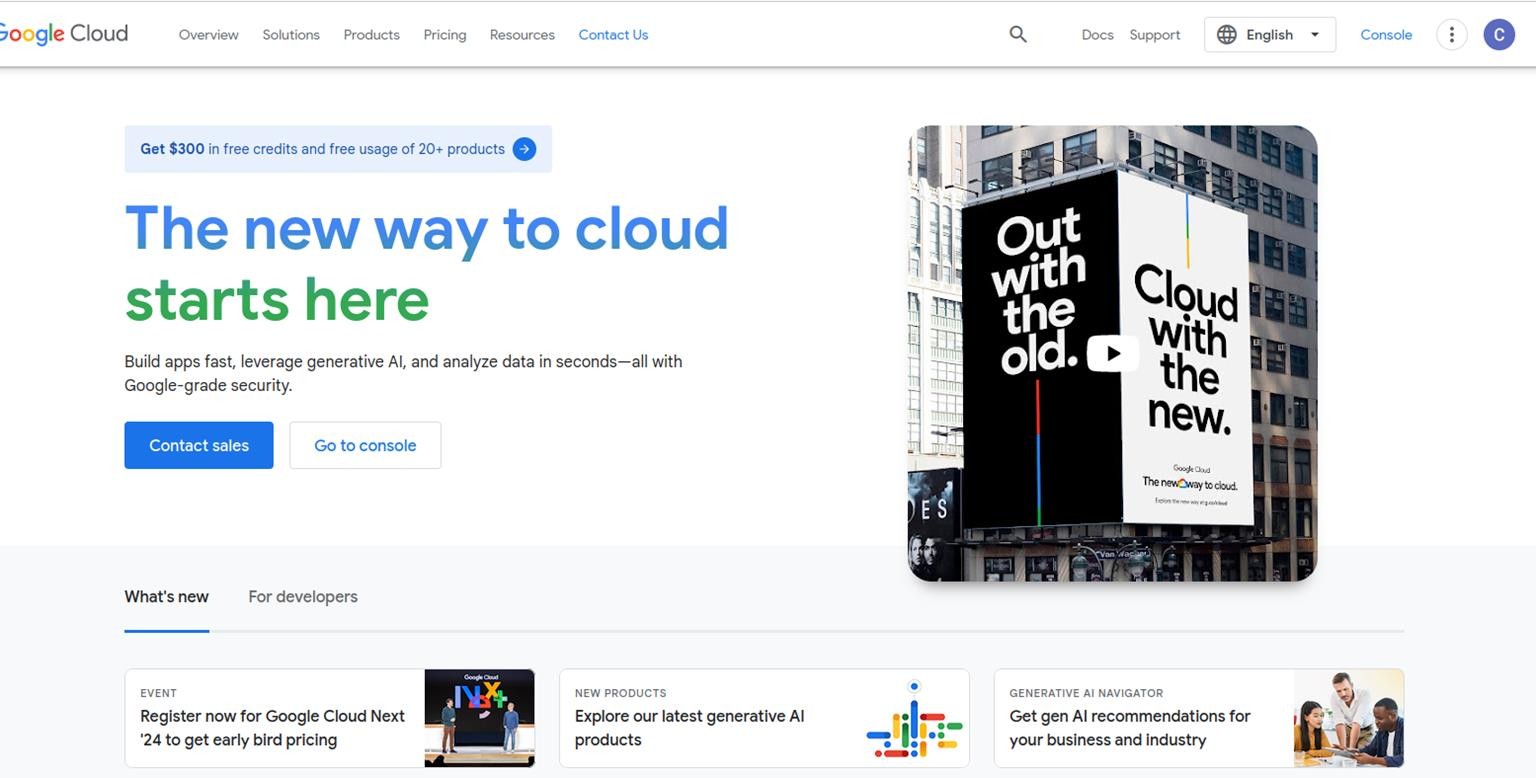
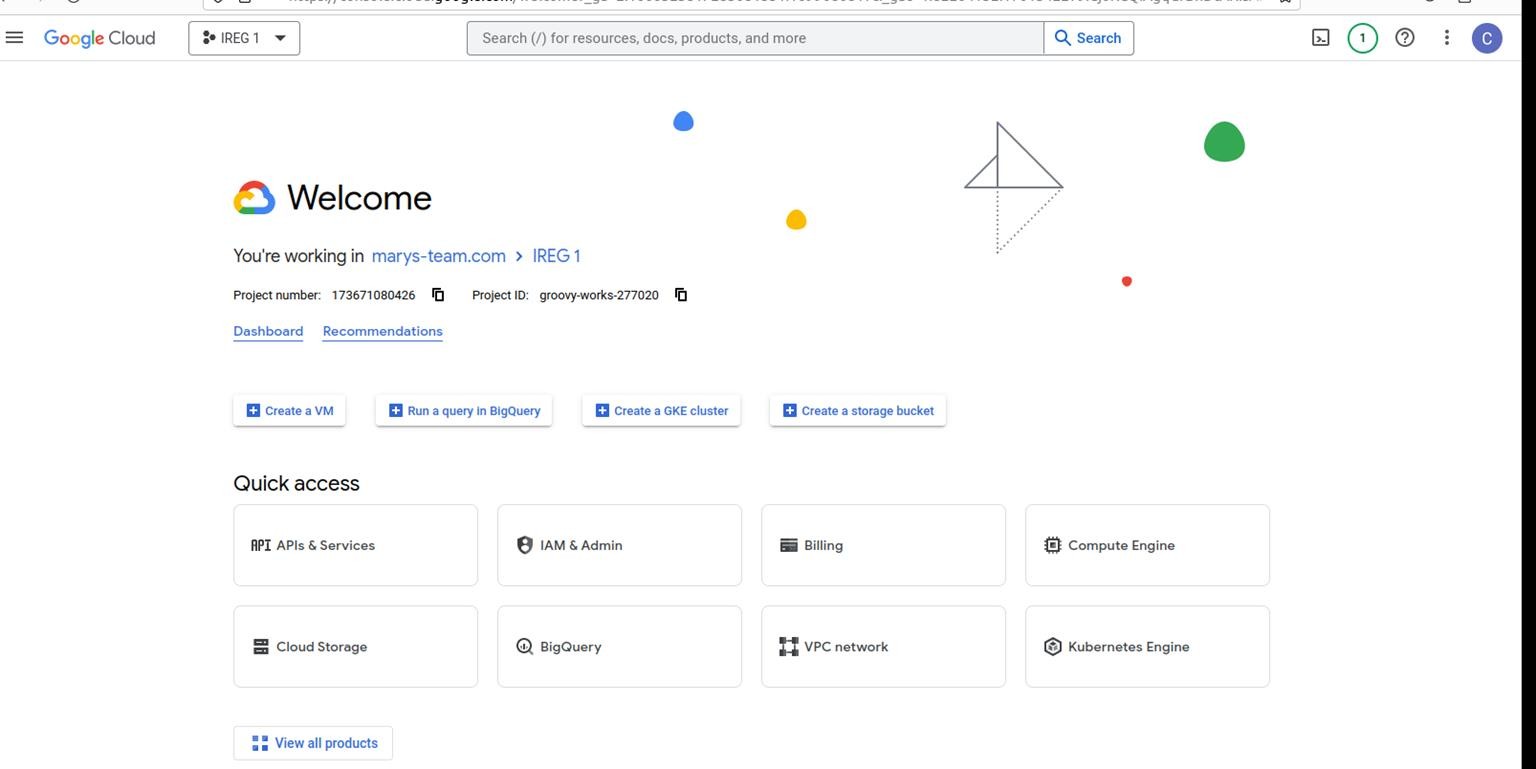
# Create a virtual machine on Google Cloud & Then Establish a Asterisk Server with Call-Centric Configuration:

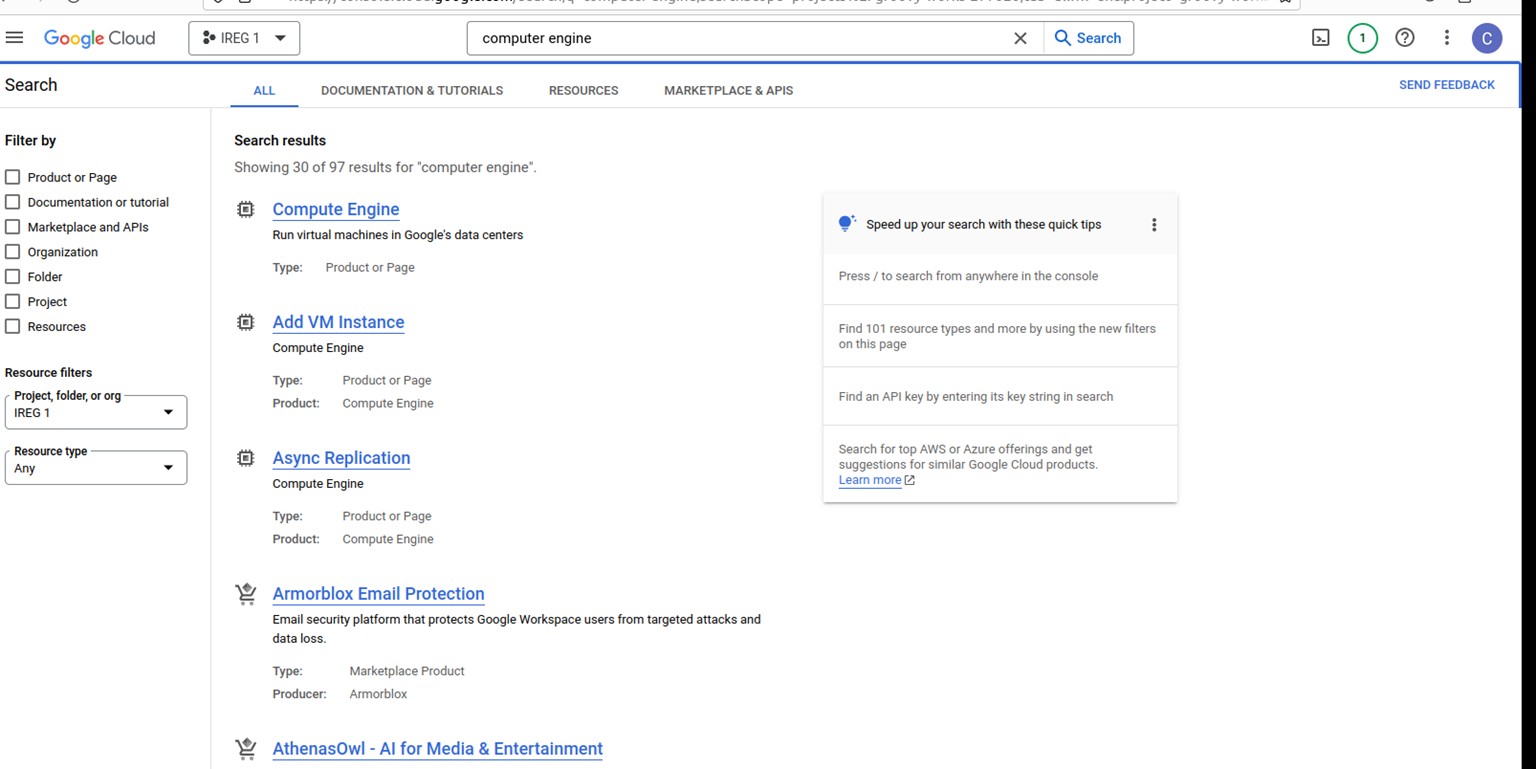
## To create a virtual machine on Google Cloud, begin by signing in to cloud.google.com/ If you sign in to Google Cloud, you will be directed to the dashboard.



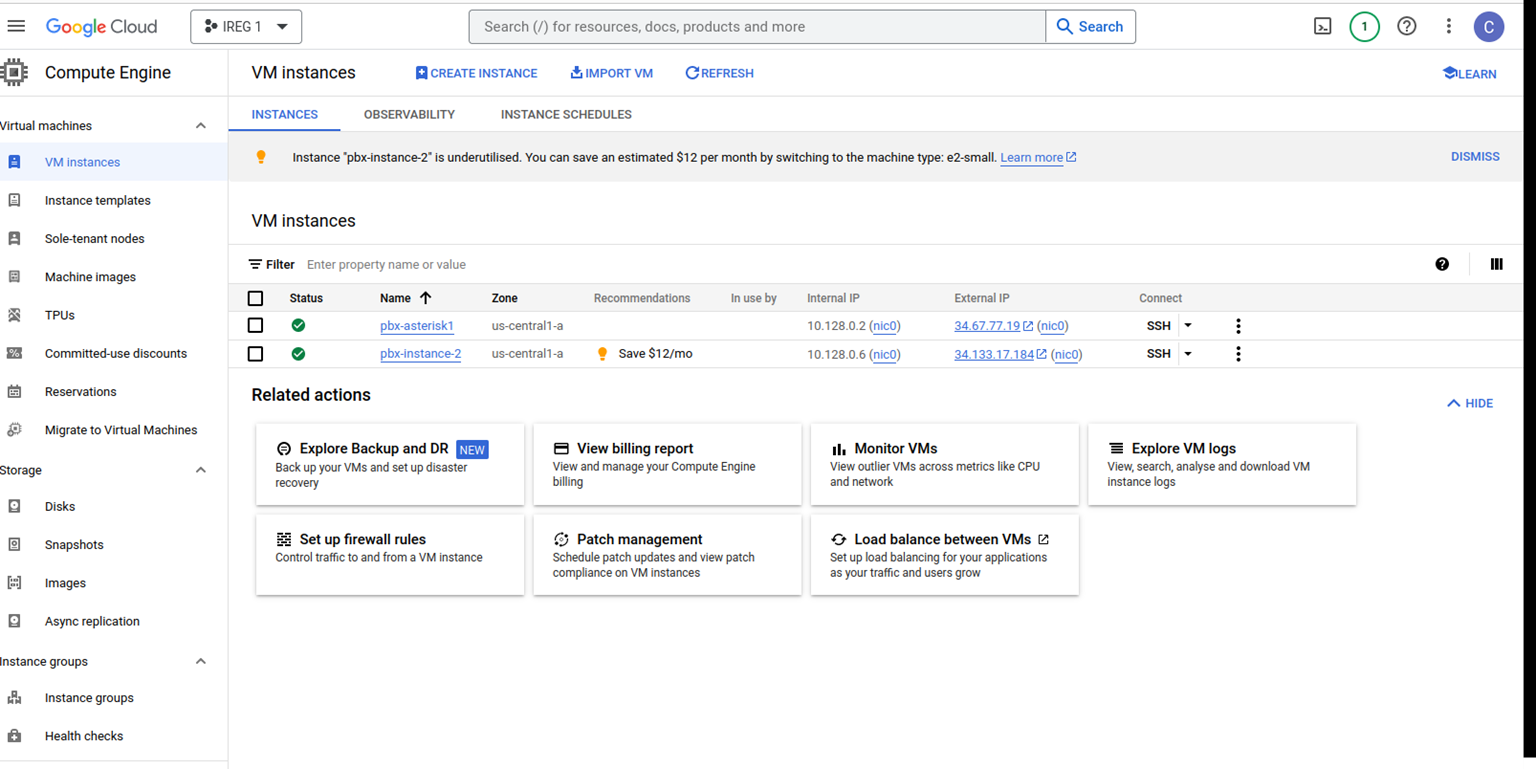
From there, click on the “Console" option located at the top of the menu.



Afterward, you will encounter this page. In the search bar at the top of the menu, enter "Compute Engine," and then press the Enter button.



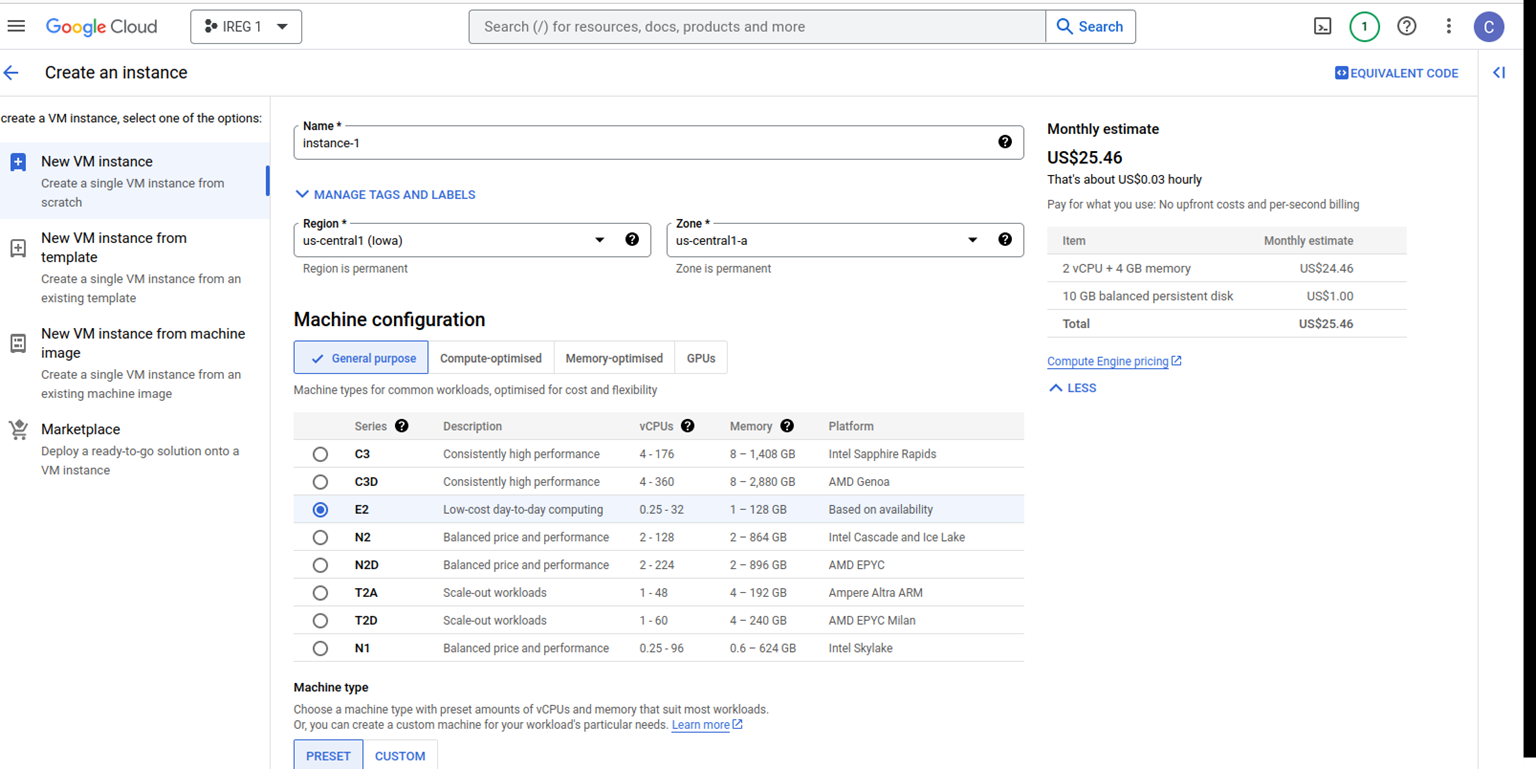
After accessing this page, click on "Compute Engine."



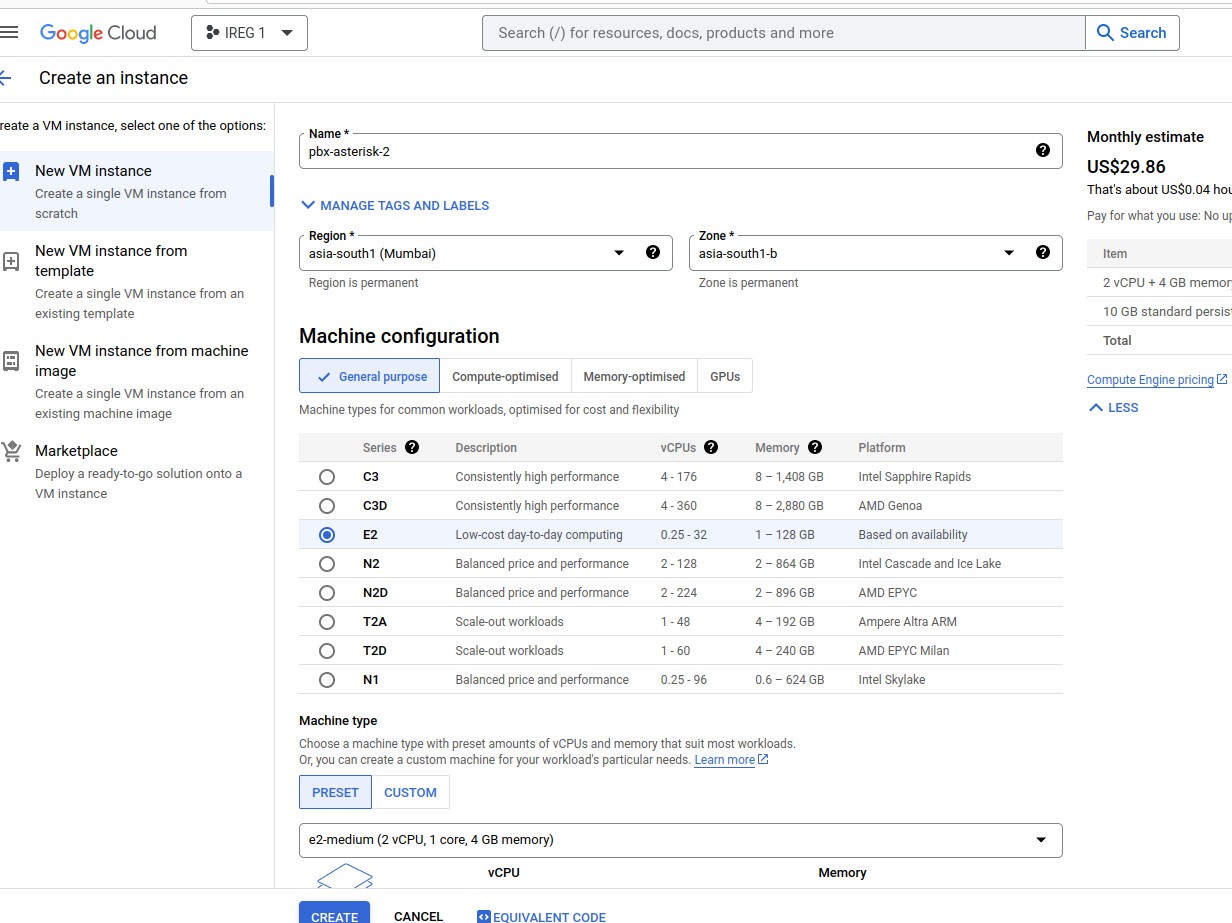
Create a VM Instance:

After accessing the Compute Engine page, click on "Create Instance" at the top of the menu. In this section, you will provide data such as:

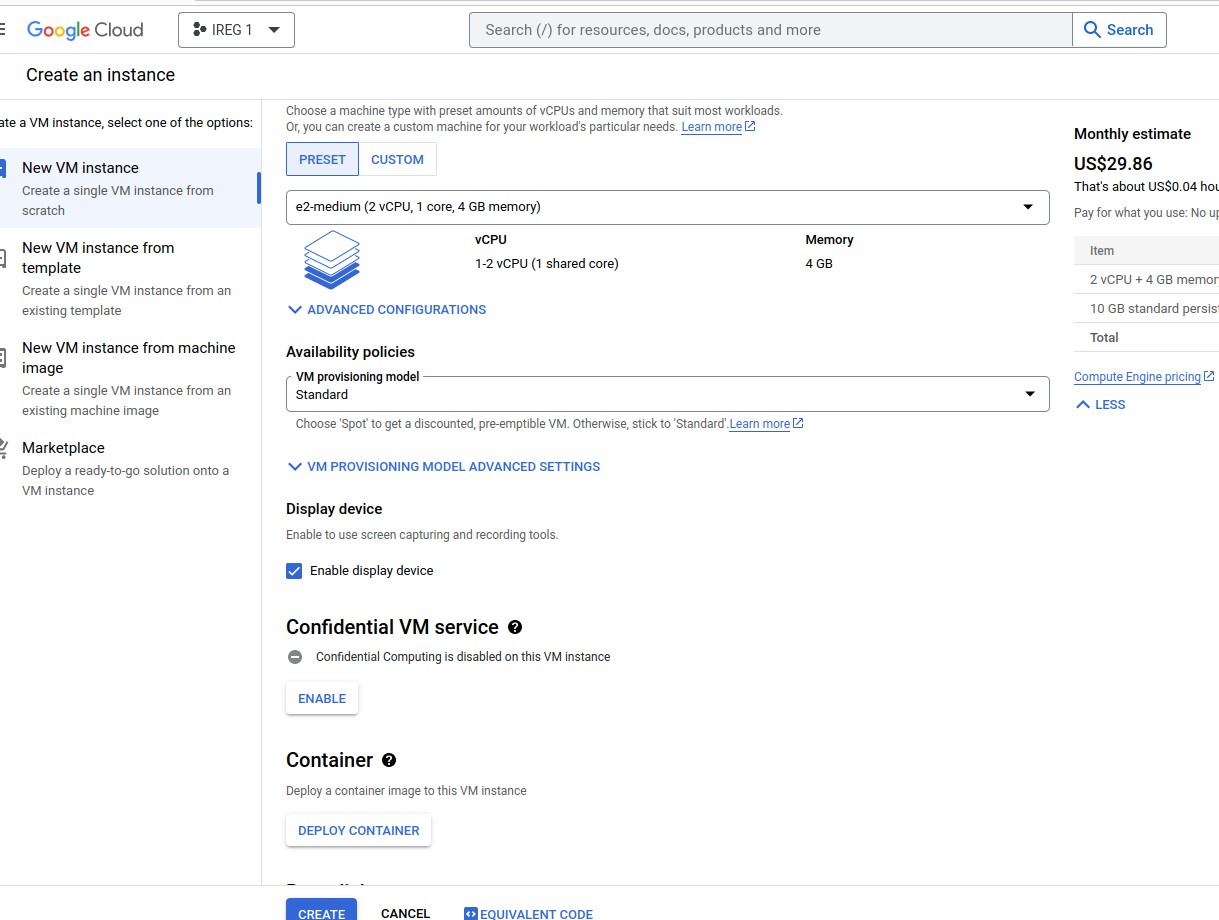
* **Name:** Enter a unique name for your virtual machine.
* **Region and Zone:** Choose the appropriate region and zone for your VM.
* **Machine type:** Specify the desired machine configuration.
* **Boot disk:** Choose the operating system for your VM.
* **Firewall:** Allow HTTP/HTTPS traffic if necessary. Follow the bellow screen shots and provide data.



## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

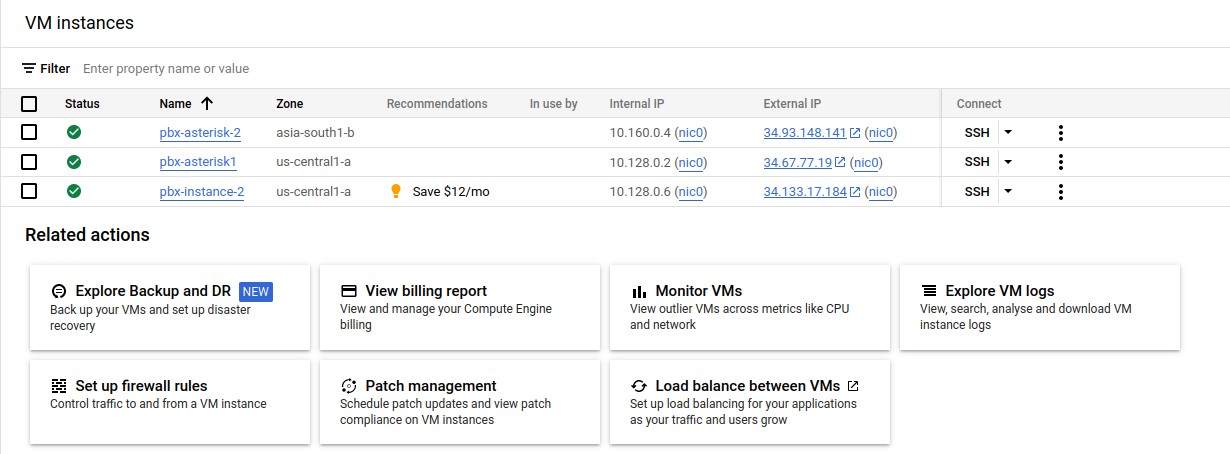


\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

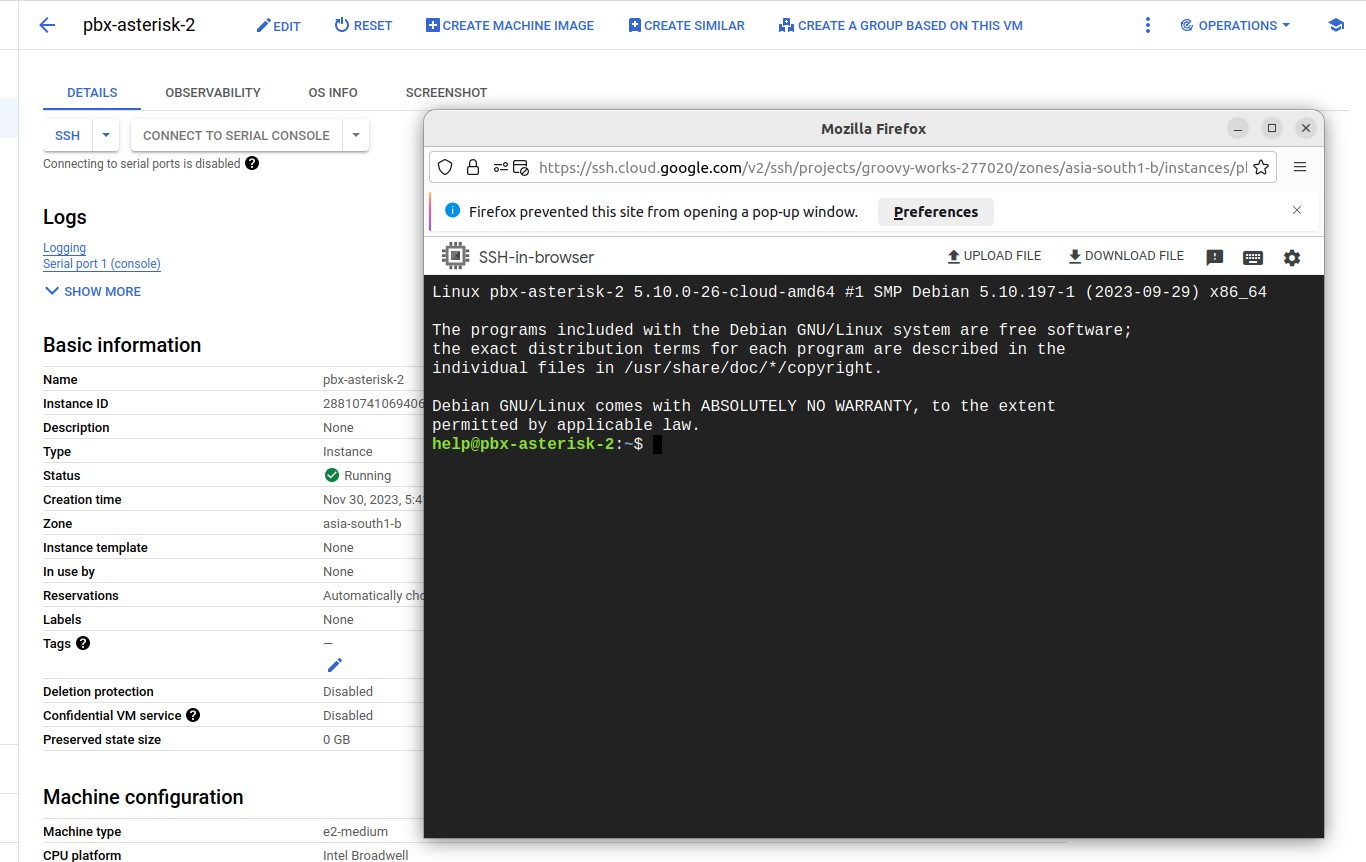


\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

After filling in the required data, you should follow the instructions provided in the above screenshots. Then, click on the "Create" button to initiate the virtual machine instance creation.



You will find the virtual machine on the dashboard, and then proceed by clicking on it.



### Connect to Your VM:

* On the VM instances page, click on the SSH button next to your VM to open a web-based SSH terminal.
* Alternatively, you can connect using an SSH client.

**Establishing a Asterisk Server with Call-Centric Configuration:**

# Step 01

**Open the [SSH-in-browser]:**

## It is advisable to use the following two commands for updating and upgrading.

1. To update the system, use the command: sudo apt-get update ;paste this command in SSH-in-browser
2. To upgrade the system, use the command: sudo apt-get upgrade ;paste this command in SSH-in-browser

**Step 02**

**To Install The Asterisk Software:**

The subsequent step involves installing the Asterisk software, an open-source communication platform, on your server. Execute the following command to accomplish this:

sudo apt-get install asterisk -y

sudo asterisk -r ;if this command run and then exit.

exit

Executing this command will install Asterisk on your server, providing you with the capability to set up and configure your local Asterisk system with Call-Centric.

**Step 03**

**Configuration:**

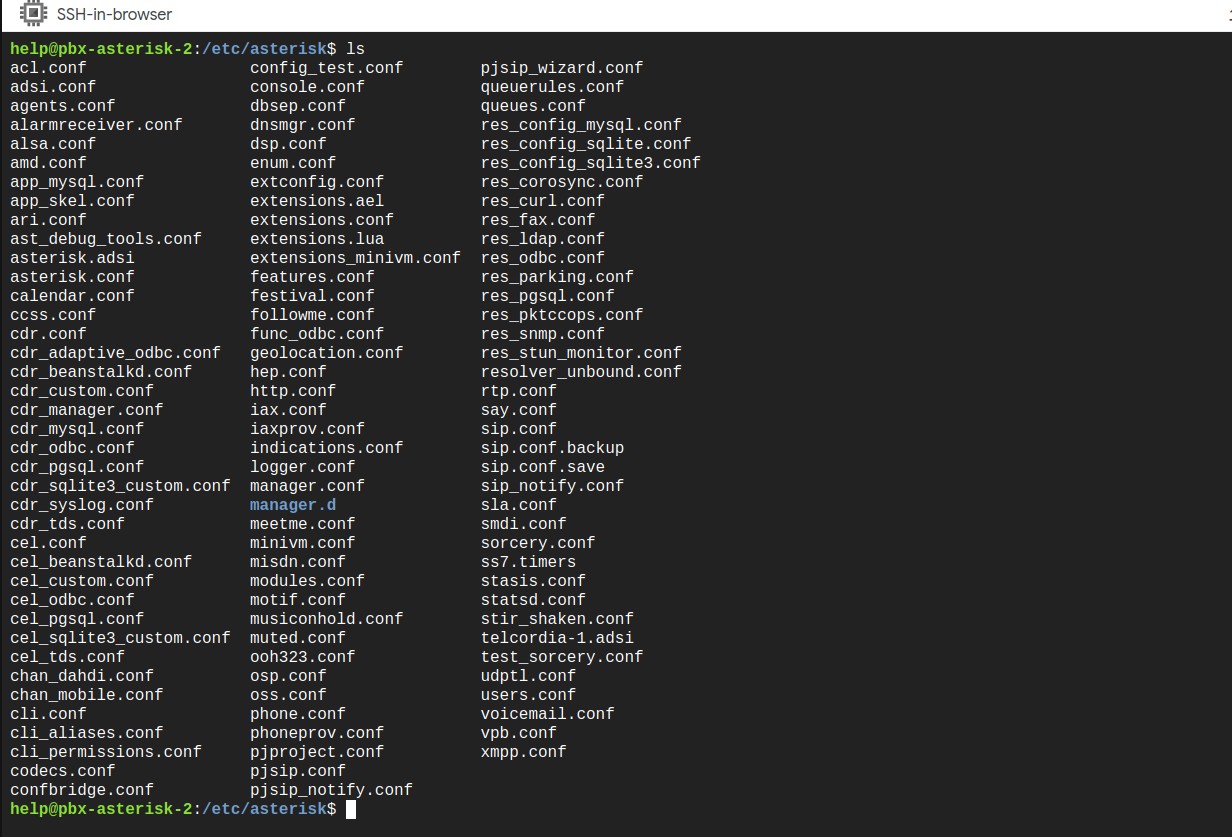
Configuring Asterisk involves modifying several .conf files to tailor them to your specific requirements. In this section, we'll guide you through the basic configuration for registering Asterisk with Call-Centric.

Navigate to Asterisk Configuration Files:

To access the Asterisk configuration files, open your terminal and use the following command:

cd /etc/asterisk/

## When you run the 'ls' command within the Asterisk directory, you will see various configuration files.



**Registering with Call-Centric**

To establish a connection with Call-Centric, configuration is done in the sip.conf file. Here's how:

1. Open the sip.conf file using the following command:
   * sudo nano sip.conf
   * Add the following code to the sip.conf file

## [general] allowguest=no

Udpbindaddr=0.0.0.0:5060 ;no changes

disallowed\_methods=UPDATE

nat = yes

externip=34.93.148.141 ;paste here external ip address, you can access from dashboard

localnet=10.160.0.0/255.255.255.0 ;paste here internal ip address, you can access from dashboard but the lass two digits will be 0.0

srvlookup=yes allowtransfers=yes dtmfmode=rfc2833 session-timers=refuse

register => Callcentric-extension-number:Password[@sip.callcentric.net](mailto:Abdullah$98765@sip.callcentric.net)

Explanation:

* + This code registers your Asterisk server with Call-Centric.
  + It includes device details, extension information, and Call-Centric credentials.
  + Below the [general] section, add the following code to define the Call-Centric configuration.

## [callcentric] type=peer

context=from-callcentric

host=sip.callcentric.net

fromdomain=sip.callcentric.net

defaultuser= Callcentric-extension-number

fromuser= Callcentric-extension-number

secret= Password

insecure=port,invite

disallowed\_methods=UPDATE

directmedia=no

videosupport=no

disallow=all allow=ulaw

[callcentric1](callcentric); host=alpha1.callcentric.com

[callcentric2](callcentric); host=alpha2.callcentric.com

[callcentric3](callcentric); host=alpha3.callcentric.com

[callcentric4](callcentric); host=alpha4.callcentric.com

[callcentric5](callcentric); host=alpha5.callcentric.com

[callcentric6](callcentric); host=alpha6.callcentric.com

[callcentric7](callcentric); host=alpha7.callcentric.com

[callcentric8](callcentric); host=alpha8.callcentric.com

[callcentric9](callcentric); host=alpha9.callcentric.com

[callcentric10](callcentric); host=alpha10.callcentric.com

[callcentric11](callcentric); host=alpha11.callcentric.com

[callcentric12](callcentric); host=alpha12.callcentric.com

[callcentric13](callcentric); host=alpha13.callcentric.com

[callcentric14](callcentric); host=alpha14.callcentric.com

[callcentric15](callcentric); host=alpha15.callcentric.com

[callcentric16](callcentric); host=alpha16.callcentric.com

[callcentric17](callcentric); host=alpha17.callcentric.com

[callcentric18](callcentric); host=alpha18.callcentric.com

[callcentric19](callcentric); host=alpha19.callcentric.com

[callcentric20](callcentric); host=alpha20.callcentric.com

[callcentricB1](callcentric); host=bravo01.callcentric.net

[callcentricB2](callcentric); host=bravo02.callcentric.net

[callcentricB3](callcentric); host=bravo03.callcentric.net

[callcentricB4](callcentric); host=bravo04.callcentric.net

[callcentricB5](callcentric); host=bravo05.callcentric.net

[callcentricB6](callcentric); host=bravo06.callcentric.net

[callcentricB7](callcentric); host=bravo07.callcentric.net

[callcentricB8](callcentric); host=bravo08.callcentric.net

[callcentricB9](callcentric); host=bravo09.callcentric.net

[callcentricB10](callcentric); host=bravo10.callcentric.net

[callcentricB11](callcentric); host=bravo11.callcentric.net

[callcentricB12](callcentric); host=bravo12.callcentric.net

[callcentricB13](callcentric); host=bravo13.callcentric.net

[callcentricB14](callcentric); host=bravo14.callcentric.net

[callcentricB15](callcentric); host=bravo15.callcentric.net

[callcentricB16](callcentric); host=bravo16.callcentric.net

[callcentricB17](callcentric); host=bravo17.callcentric.net

[callcentricB18](callcentric); host=bravo18.callcentric.net

[callcentricB19](callcentric); host=bravo19.callcentric.net

[callcentricB20](callcentric); host=bravo20.callcentric.net

Explanation:

* This code specifies the configuration parameters for Call-Centric.
* It defines the connection details and security settings.
* You can also define multiple Call-Centric configurations using similar blocks like [callcentric1], [callcentric2], and so on.

# Step 04

**Asterisk Dial Plan:**

## Following the registration, we create a dial plan for the server by adding the following code to the **extensions.conf** file

[from-callcentric]

(1) exten => s,1,Set(Var\_TO=${CUT(CUT(SIP\_HEADER(To),@,1),:,2)})

This code extracts the 'To' header information from the SIP message and stores it in the Var\_TO variable.

(2) exten => s,n,GotoIf($["${Var\_TO}" = " Callcentric-extension-number"]?internal,s,1:3)

It checks if the Var\_TO variable matches "17778519255108" and routes to the 'internal' context if true, otherwise proceeds to priority 3.

(3) exten => s,n,GotoIf($["${Var\_TO}" = "Dial-ID(12033873156)"]?internal,s,1:4)

Similar to the previous line, this one checks if Var\_TO matches "12033873156" and routes accordingly.

(4) exten => s,n,Hangup()

If none of the conditions match, it ends the call. [internal]

(1) exten => s,1,Goto(demo-menu,s,1)

This line directs the call to the 'demo-menu' context.

## [demo-menu]

This context handles the main menu options:

1. exten => s,1,Answer(500)

## Answers the call with a 500ms delay.

1. same => n(loop),Background(welcome-ireg)

## Plays the 'Main menu' sound file in a loop but the file must be in wav formate.

1. same => n,WaitExten(5)

## Waits for user input for 5 seconds.

1. exten => 1,1,WaitExten(2)

## Waits for user input for 2 seconds, typically for the English menu.

1. same => n,Goto(english-menu,s,1)

## Routes to the 'english-menu' context if '1' is pressed.

1. exten => 2,1,WaitExten(2)

## Waits for user input for 2 seconds, typically for the Spanish menu.

1. same => n,Goto(spanish-menu,s,1)

## Routes to the 'spanish-menu' context if '2' is pressed.

1. exten => i,1,Playback(option-is-invalid)

## Informs the user that the option invalid.

1. same => n,Goto(s,loop)

Returns to the main menu loop.

1. exten => t,1,Playback(are-you-still-there)

Asks the user if they are still there.

1. same => n,Goto(s,loop)

Returns to the main menu loop.

[english-menu]

exten => s,1,Answer(500)

same => n,Set(GLOBAL(Extension)=1)

same => n,Set(GLOBAL(ExtensionsList)=)

same => n(loop),Background(task1)  
 ;task1 is the sound but that file will be in wav formate.

same => n,WaitExten(5)

exten => 1,1,WaitExten(1)

same => n,Set(GLOBAL(Extension,1)=100)

same => n,Set(GLOBAL(ExtensionsList)=${GLOBAL(ExtensionsList)}:Extension,1=${GLOBAL(Extension,1)})

same => n,Goto(to-100,s,1)

exten => 2,1,WaitExten(2)

same => n,Set(GLOBAL(Extension,2)=101)

same => n,Set(GLOBAL(ExtensionsList)=${GLOBAL(ExtensionsList)}:Extension,2=${GLOBAL(Extension,2)})

same => n,Goto(to-101,s,1)

exten => 0,1,WaitExten(2)

same => n,Goto(demo-menu,s,1)

exten => i,1,Playback(option-is-invalid)

same => n,Goto(s,loop)

exten => t,1,Playback(are-you-still-there)

same => n,Goto(s,loop)

;This context handles the Spanish menu options:

[spanish-menu]

exten => s,1,Answer(500)

same => n(loop),Background(spanish)

same => n,WaitExten(5)

exten => 1,1,WaitExten(1)

same => n,Goto(to-100,s,1)

exten => 2,1,WaitExten(2)

same => n,Goto(to-101,s,1)

exten => 0,1,WaitExten(2) ; go back

same => n,Goto(demo-menu,s,1)

exten => i,1,Playback(option-is-invalid)

same => n,Goto(s,loop)

exten => t,1,Playback(are-you-still-there)

same => n,Goto(s,loop)

;(1) Similar to the 'demo-menu', it handles menu options in Spanish.

## [to-phone-number 1]

This context handles calls directed to extension 'phone-number 1':

(1) exten => s,1,Dial(SIP/phone-number 1,20):

Initiates a call to extension 'phone-number 1' with a timeout of 20 seconds.

(1) exten => s,n,Hangup:

[to-phone-number 2]

This context handles calls directed to extension '101':

1. exten => n,Dial(SIP/phone-number 2,20,m):

Initiates a call to extension 'phone-number 2' with a timeout of 20 seconds.

1. exten => n,Hangup:

Hangs up the call after the Dial command completes.

## [to-callcentric]

This context allows dialing external numbers:

(1) exten => \_X.,1,Dial(SIP/${EXTEN}@callcentric):

Allows dialing any external number starting with any digit. It routes the call using the specified SIP trunk.

# Step 05

**Asterisk Call Recording:**

### : Create a Recording Directory

To store recorded calls, create a dedicated directory within Asterisk: sudo mkdir /var/lib/asterisk/recordings

### : Check Directory Ownership

It's crucial to ensure that the directory is owned by the Asterisk user and group. You can verify ownership using the following command:

ls -ld /var/lib/asterisk/recordings

If the output appears like this:

drwxrwxr-x 2 asterisk asterisk 4096 August 10 14:00 /var/lib/asterisk/recordings Proceed to step 3. If not, move to step 2.1.

### 2.1 : Change Directory Ownership

To change ownership to the Asterisk user and group, execute:

sudo chown -R asterisk:asterisk /var/lib/asterisk/recordings

### : Grant Write Permissions

Grant write permissions to both the owner and group with the following command: sudo chmod 775 /var/lib/asterisk/recordings

### : Restart Asterisk

To apply the changes, restart the Asterisk service:

sudo systemctl restart asterisk

### : Configure Call Recording in extensions.conf

In your extensions.conf file, insert the following code to initiate call recording: Plaintext

This code will create a recording file in WAV format, timestamped with the current date and time. The recorded calls will be stored in the /var/lib/asterisk/recordings directory.

**Step 06**

**Music On Hold:**

**Choose or Create Music Files:**

First, you need to have audio files that you want to use for MOH. These can be music tracks, announcements, or any audio you want to play to callers while they are on hold. Ensure these audio files are in a compatible format like WAV.

### : Configure MOH Class:

In Asterisk, you define MOH classes, which group audio files together. Each class can have multiple audio sources. You can define MOH classes in the musiconhold.conf configuration file.

Here's an example:

[default] mode=files

directory=/var/lib/asterisk/moh

In this example, we create a class named "default" that plays audio files located in the

/usr/share/asterisk/moh directory.

### : Assign MOH Class to Calls:

In your dial plan (usually in extensions.conf), you can specify which MOH class to use for specific calls.

For example:

[to-phone-number 1]

exten => s,1,Set(MOHCLASS=default) ; Set MOH class to "default" same => n,Dial(SIP/phone-number 1,20,m) ; Use MOH for the call

In this example, we set the MOH class to "default" for the call to extension 100.

### : Enable MOH for Calls:

Ensure that MOH is enabled for your calls. This can be done using the m option in the Dial command, as shown in the previous example. The m option instructs Asterisk to play MOH for this call.

# Step 07

**Call Transfers:**

## Asterisk offers two primary methods for call forwarding: blind transfer and attended transfer. While blind transfer swiftly redirects calls without intervention, we strongly advocate for the implementation of attended transfer in call forwarding scenarios. This method ensures a superior level of security compared to blind transfer, which may pose a risk of misuse due to its more immediate nature.

**Configuration Steps:**

### Enable Attended Transfer in features.conf:

To enable attended transfer, locate and edit the features.conf file in your Asterisk configuration. Uncomment the "atxfer" line as follows:

[featuremap]

;blindxfer => #1

;disconnect => \*0

;automon=1 atxfer => 2

;parkcall => #72

;automixmon => \*3

Update extensions.conf:

Next, you need to incorporate the following lines of code into your extensions.conf file:

exten=>s,1,Dial(SIPphone-number 1 ,20,t)

exten => s,n,Hangup()

exten =>77,1,Goto(tophone-number 2,s,1)

# 

# Transfer Process:

### : User Request for Transfer:

When a user on extension phone-number 1 wishes to transfer a call, they simply need to inform the agent on extension phone-number 2 about their intention to transfer.

### : User Prompt:

The agent on extension phone-number 1 can then instruct the user to initiate the transfer process by pressing the number '2' on the Dual-Tone Multi-Frequency (DTMF) keypad of their phone.

### : Initiating the Transfer:

After the user presses '2', the system is configured to recognize this action as a request for transfer.

### : Agent's Role:

The agent, upon receiving the transfer request, can then dial '77' on their DTMF keypad. This initiates the call transfer process.

### : Establishing the Connection:

Once '77' is dialed, the system establishes a connection between the user and the target extension (in this case, extension phone-number 2).

### : Ongoing Communication:

### The user on extension phone-number 1 is now connected to extension phone-number 2 and both parties can communicate as needed.

### : Call Termination:

When the conversation is complete, either party can hang up their phone. If extension phone- number 2 hangs up, the channel between extension 101 and the user is maintained, ensuring the call can continue.

# Step 8 Api For sending Data to clickflo website:

API for transferring data from Asterisk to a website. The API is built using Python and allows for customized data transfer. Follow these steps to set up the API.

### Step 1: Install Python 3

Ensure that Python 3 is installed on your system. You can download and install it from the official Python website (<https://www.python.org/downloads/>).

### Step 2: Create the Python Script

1: Create a Python script file in the /etc/asterisk/ directory. You can use any text editor to create this file. Name it, for example, api\_request.py.

2: Paste the following Python code into the api\_request.py file:

#!/usr/bin/env python3 import requests

import sys

API\_URL = "https://clicflo.com/api/1.1/wf/send\_sms\_3cx" API\_KEY = "5b68a7e19a0b38394852b5600c81f45e"

def send\_sms\_3cx(CALLER\_ID\_NUM, SELECTED\_EXTENSION): payload = {

"CALLER\_ID\_NUM": CALLER\_ID\_NUM, "SELECTED\_EXTENSION": SELECTED\_EXTENSION

}

headers = {

"Authorization": f"Bearer {API\_KEY}", "Content-Type": "application/json"

}

response = requests.post(API\_URL, json=payload, headers=headers) return response.status\_code, response.text

if name == " main ": if len(sys.argv) != 3:

print("Usage: python3 api\_request.py CALLER\_ID\_NUM SELECTED\_EXTENSION")

sys.exit(1)

CALLER\_ID\_NUM = sys.argv[1] SELECTED\_EXTENSION = sys.argv[2]

status\_code, response\_text = send\_sms\_3cx(CALLER\_ID\_NUM, SELECTED\_EXTENSION)

print(status\_code) print(response\_text)

### Step 3: Configure in the Dial Plan

Add the following code to your Asterisk extension configuration. Replace

/etc/asterisk/api\_request.py with the actual path to your Python script:

## exten => s,2,NoOp(Starting API script)

same => n,Set(CALLER\_ID\_NUM=${CALLERID(num)})

same => n,Set(SELECTED\_EXTENSION=${SELECTED\_EXTENSION})

same => n,Set(API\_SCRIPT=/etc/asterisk/api\_request.py) ; Replace with your actual path same =>

n,AGI(${API\_SCRIPT},${CALLER\_ID\_NUM},${SELECTED\_EXTENSION})

same => n,NoOp(API request completed)

**Step 09 Establishing a Feedback System:**

To establish a feedback system in your Asterisk configuration, follow these steps:

**Step 1: Prepare Directory Structure**

Begin by creating a directory named "feedback" within /var/lib/asterisk/. This directory will house your feedback data. Ensure that this directory is accessible by Asterisk and write permissions are granted to it.

## Mkdir /var/lib/asterisk/feedback chmod 775 /var/lib/asterisk/feedback

**Step 2: Create a CSV File**

Inside the "feedback" directory, create a CSV file named feedback.csv. This file will store your feedback data.

touch /var/lib/asterisk/feedback/feedback.csv chmod 664 /var/lib/asterisk/feedback/feedback.csv **Step 3: Configure Your Dial Plan**

Add the following code to your extensions.conf file to configure your dial plan:

## [Questions]

exten => 1,1,Playback(question1)

same => n,Set(DID\_NUMBER=${Var\_TO}) same => n,Read(response1,beep,1)

same => n,Verbose(Response to Question 1 is ${response1})

same => n,Set(CSV\_LINE="TimeStamp:

${STRFTIME(${EPOCH},,%Y-%m-%d %H:%M:%S)},DID Number:

${DID\_NUMBER},CallerId: ${CALLERID(num)} ${Extension}")

same => n,System(echo "${CSV\_LINE}" >> /var/lib/asterisk/feedback/feedback.csv) same => n,Set(CSV\_LINE2="Question 1: How satisfied are you with our product or

service?,${response1} Star")

same => n,System(echo "${CSV\_LINE2}" >> /var/lib/asterisk/feedback/feedback.csv) same => n,Goto(2,1)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* exten => 2,1,Playback(question2)

same => n,Read(response2,beep,1)

same => n,Verbose(Response to Question 2 is ${response2})

same => n,Set(CSV\_LINE="Question 2: How likely are you to recommend us to a friend or colleague?,${response2} Star")

same => n,System(echo "${CSV\_LINE}" >> /var/lib/asterisk/feedback/feedback.csv) same => n,Goto(3,1)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* exten => 3,1,Playback(question3)

same => n,Read(response3,beep,1)

same => n,Set(CSV\_LINE="Question 3: How would you rate the quality of our customer support?,${response3} Star")

same => n,System(echo "${CSV\_LINE}" >> /var/lib/asterisk/feedback/feedback.csv)

same => n,Goto(4,1)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* exten => 4,1,Playback(question4)

same => n,Read(response4,beep,1)

same => n,Set(CSV\_LINE="Question 4: Did our product or service meet your expectations?,${response4} Star")

same => n,System(echo "${CSV\_LINE}" >> /var/lib/asterisk/feedback/feedback.csv) same => n,Goto(5,1)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* exten => 5,1,Playback(question5)

same => n,Read(response5,beep,1)

same => n,Set(CSV\_LINE="Question 5: Rate the level of customer service you received?,${response5} Star")

same => n,System(echo "${CSV\_LINE}" >> /var/lib/asterisk/feedback/feedback.csv) same => n,Playback(thanks)

same =>

n,Set(CurrentDateTime=${STRFTIME(${EPOCH},,%Y-%m-%d %H:%M:%S)}) same => n,Set(DID\_NUMBER=${Var\_TO})

same => n,Set(CALLER\_ID\_NUM=${CALLERID(num)}) same => n,Set(SELECTED\_EXTENSION=${Extension}) same => n,Set(API\_SCRIPT=/etc/asterisk/api\_request.py)

same => n,AGI(${API\_SCRIPT},${CurrentDateTime},${DID\_NUMBER},${CALLER\_ID\_NUM},

${S ELECTED\_EXTENSION})

same => n,Hangup()

**Step 4: Include Configuration in Dial Plan**

In the relevant sections of your extensions.conf file (e.g., "to-phone-number 1" and "to- phone-number 1"), include the following code:

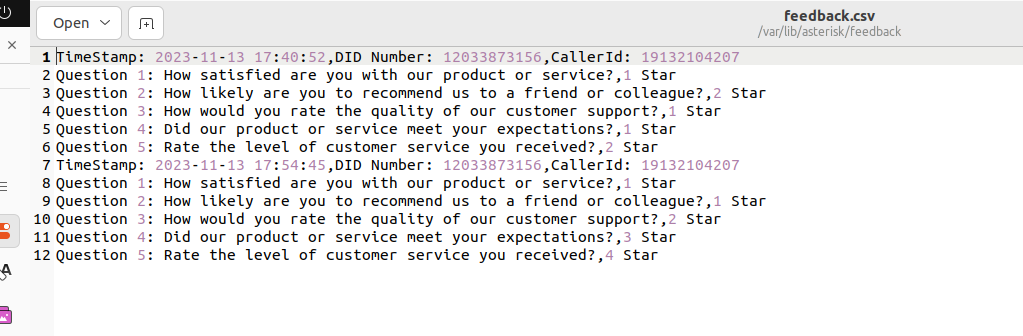
exten => 78,1,Goto(feedback,s,1)

## [feedback]

exten => s,1,Background(feedback) same => n,Goto(Questions,1,1) same => n,Hangup()

With this setup, you will collect feedback responses and recls ord them in the CSV file located in /var/lib/asterisk/feedback/feedback.csv.

**Output:**



# Step 10

# Simultaneously call to five numbers:

## [call-multiple]

exten => s,1,Answer()

same => n,Dial(SIP/callcentric/+15168471031$SIP/callcentric/+3456879483$SIP/callcentric/+34598 76423,30)

Add the above code to the extensions.conf file.

**Step 11**

**Asterisk call status:**

### BUSY: The called party is currently busy with another call.

**NOANSWER:** The called party did not answer the call.

**ANSWER:** The called party answered the call.

**CANCEL:** The calling party canceled the call attempt before it was answered.

**CHANUNAVAIL:** The requested channel is not available for the call.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

exten => h,1,Set(CallResult=${DIALSTATUS}) same => n,Noop(Result: ${CallResult})

same => n,Hangup()

For the status you should have to put the above code in the extensions.conf file with step 10 code