3rd party services used by Focus server

These suppliers mostly offer the capability to regionalise their services.

So far as possible, for any one server, configure services to regions in the same jurisdiction for every supplier.

NB: there is some configuration information for SIP calls out of a Twilio conference call into the video meeting still to be provided.

The 3rd party suppliers are:

|  |  |  |
| --- | --- | --- |
| **Supplier** | **Service(s)** | **Web site** |
| Vonage | Video API |  |
| Amazon Web Services | Storage for recordings |  |
| Twilio | Telephony |  |
| Google | Transcription and translation |  |
| Temi | Offline transcription |  |

# Twilio

* Set up a Twilio account (twilio.com)
* With the console, you will need to configure the Voice, Chat and Sync products

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Voice

* Configure the web hook in your Voice configuration for the /sayhello endpoint on your domain
* You will need the Account SID and Auth Token for your settings\_local.py
* For each phone number you want to associate with your server:
  + Adapt the TWILIO\_PHONE\_NUMBERS JSON with your phone number and a suitable display name for it. If you specify a non-UK phone number or wish to use a different voice for the IVR announcements you should refer to the Twilio documentation for suitable values for ‘voice’ and ‘language’

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

## Chat configuration

* Configure a chat service with its webhooks for your domain
* You will need the service sid and chosen friendly name for your app.env

A screenshot of a computer

Description automatically generated

# A screenshot of a computer Description automatically generated

## SIP Domain: You will need to set up a SIP domain for your project:A screenshot of a computer Description automatically generated

Click + to add your new domain with your own chosen friendly name and domain name component:

A screenshot of a computer

Description automatically generated

This is an example for the present nvision server. You will need a credential list so the screenshot shows the fields you have to provide when you click to add a credential list.

Use your own server domain in the web hook URL

A screenshot of a computer

Description automatically generated

Secure Media must be enabled:

A screenshot of a computer

Description automatically generated

Use the information entered and displayed on these screens to populate these variables in app.env:

* TWILIO\_SIP\_DOMAIN
* TWILIO\_SIP\_USERNAME
* TWILIO\_SIP\_PASSWORD
* TWILIO\_SIP\_SECURE = True

## Sync service

You will need to create a Sync service with your chosen friendly name and identify it in app.env:

A screenshot of a computer

Description automatically generated

# Vonage

* Create a Vonage Video API account
* Within the account create a project with a name of your choice to associate with the server
* You will need the project name, API key and secret for your settings\_local.py
* Configure the two webhooks for your custom domain (same endpoints, but your domain name)
* Configure your archiving storage provider. Presently this must be AWS or at least S3 compatible because the use of S3 protocols to retrieve recordings is hardcoded in the application.

A screenshot of a login screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

NB: Video codec choice may need to be reconsidered. VP8 is now recommended.

A screenshot of a computer

Description automatically generated

Editing cloud storage settings:

A screenshot of a computer

Description automatically generated

The access keys and bucket name are from your AWS setup (below).

# AWS

AWS buckets are used for archival storage and for static web assets.

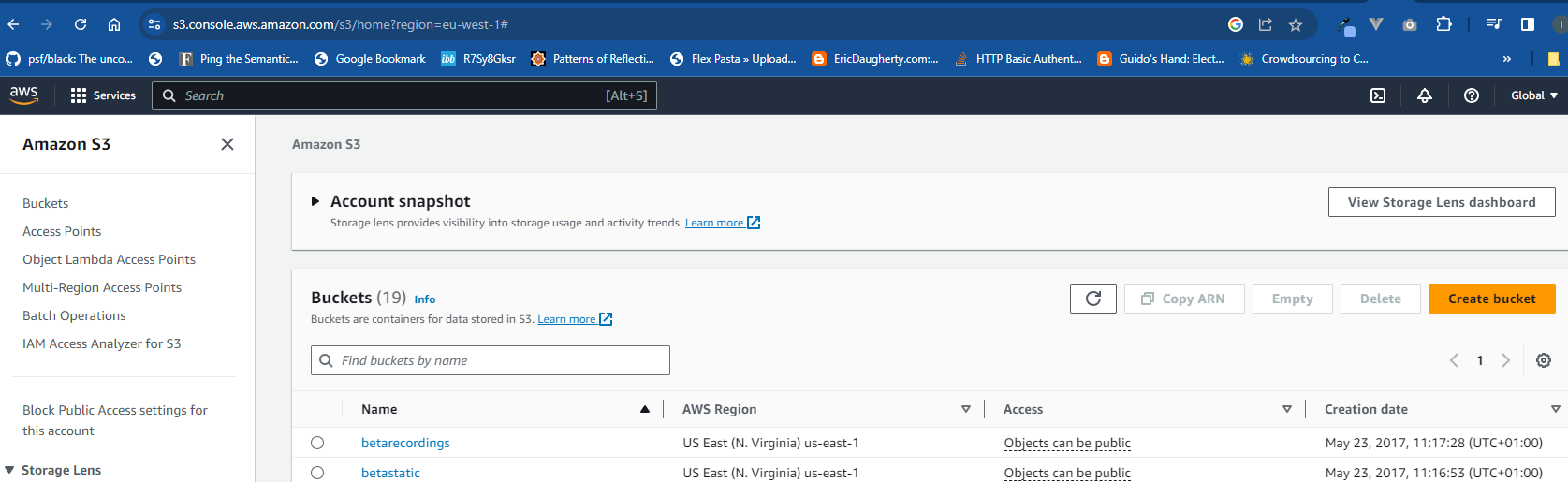
You require two buckets, one for static assets and another for recordings.

The recordings bucket should be managed by one AWS IAM user used to configure the Vonage API – the AWS credentials for that user go into the Vonage portal as above, and the assets buckets may be managed by another AWS IAM user identified in app.env. For simplicity you can use a single user and key/secret pair.

Generate an API key and secret for each user

You will require the following information from AWS for your app.env:

* AWS\_API\_KEY
* AWS\_SECRET
* AWS\_REGION\_CODE
* AWS\_REGION\_NAME



This explains the process of creating the account and the necessary users: <https://chat.openai.com/share/4c45f7f6-0fad-4a18-ba23-65348cfb6fc9>

You don’t need to be concerned with the information about signed URLs just to configure and operate the system – it’s just FYI.

Note that the recording bucket requires some CORS settings in its Permissions:

A screenshot of a computer

Description automatically generated

This is the JSON text to paste into your recording bucket CORS settings:

[

{

"AllowedHeaders": [],

"AllowedMethods": [

"GET"

],

"AllowedOrigins": [

"\*"

],

"ExposeHeaders": []

}

]

# Google

You require a Google project and a JSON file containing the necessary Google API credentials for the project.

The project is identified in your settings with the symbol GOOGLE\_PROJECT\_ID.

The region for the project should be in REGION\_ID.

Set up the Google account (if necessary) and project following this guidance https://chat.openai.com/share/ac3f30fe-5dcb-4f3d-abc8-43397b591060

The JSON file created by Google should be stored as the file /var/app/focus-distrib/gcs\_speech\_api.json

# Temi

Temi is an offline speech transcription API. Set up an account at temi.com and create an API key on their developer page at https://www.temi.com/account/developer

Use the key as your value for TEMI\_API\_KEY in app.env