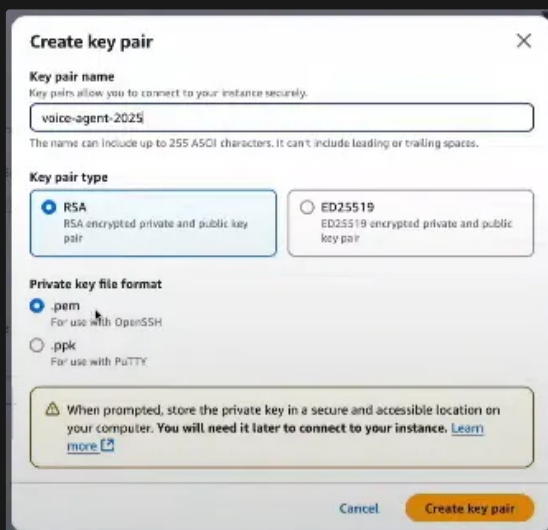


黑客松 - 語音助理

EC2

1. 查詢 ec2 - lunch instance
2. 命名 voice - agent
3. 作業系統一律選 ubuntu -
 - a. 22.04 的第二個
 - b. instance type - t3 xlarge
 - c. (不一定要 key)



The screenshot shows the 'Create key pair' dialog box in AWS. It has a title bar with a close button (X). The main content area is divided into sections: 'Key pair name' with a text input field containing 'voice-agent-2025', 'Key pair type' with two radio button options 'RSA' (selected) and 'ED25519', and 'Private key file format' with two radio button options '.pem' (selected) and '.ppk'. A warning box at the bottom states: 'When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. Learn more'. At the bottom right are 'Cancel' and 'Create key pair' buttons.

Create key pair [X]

Key pair name
Key pairs allow you to connect to your instance securely.
voice-agent-2025
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ **RSA**
RSA encrypted private and public key pair

☐ **ED25519**
ED25519 encrypted private and public key pair

Private key file format

☒ **.pem**
For use with OpenSSH

☐ **.ppk**
For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Cancel Create key pair

4. Network setting
 - a. create security group
 - b. allow SSH
 - c. 右上角 edit 點下去
 - d. add security group rule
 - e. 項目

The image shows two screenshots of the AWS Management Console's Security Group Rule configuration page. The top screenshot is for 'Security group rule 2 (TCP, 8080, 0.0.0.0/0)' and the bottom is for 'Security group rule 3 (TCP, 3000, 0.0.0.0/0)'. Both rules are configured with 'Type: Custom TCP', 'Protocol: TCP', 'Port range: 8080' (top) and '3000' (bottom), 'Source type: Custom', 'Source: 0.0.0.0/0', and 'Description: e.g. SSH for admin desktop'. A warning message is visible in the top screenshot: 'Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.'

5. 項目

The image shows the 'Configure storage' page in the AWS Management Console. It displays '1x 30 GiB gp2' storage configuration. A note states: 'Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage'. There is an 'Add new volume' button and a warning: 'The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance.'

6. 右邊確認 model - launch instance
7. 回到上一動確認 instance

The image shows the 'Instances' page in the AWS Management Console. It displays a table with two instances:

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	my-voice-agen...	i-0b5301b46cdf74910	Running	t3.medium	3/3 checks passed	View alarms +	us-west-2c
<input type="checkbox"/>	voice-agent	i-03412991f43bf52a8	Running	t3.xlarge	3/3 checks passed	View alarms +	us-west-2a

8. 點進剛剛建立好的 ec2 - 進去後右邊點 connect - 再進去之後再進去 connect - 就可以連線進入 terminal

從這邊之後就去建立 CloudFront

以下是建立玩 CloudFront 之後才做的事情

以下是環境的安裝

- 回到 terminal

複製文件裡面的 Docker

```
all Docker & Docker Compose

Docker: https://docs.docker.com/engine/install/ubuntu/
Docker Compose: https://docs.docker.com/compose/install/linux/

The installation commands are as follows:

Configure apt source

# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
1 "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
2 ${. /etc/os-release && echo "VERSION_CODENAME"} stable" | \
3 sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

貼上 terminal 之後按 enter

```
all the latest Docker

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

再貼上後按 enter，中間 y/n 要記得按 y

```
try Docker installation

sudo docker run hello-world
```

看到 Hello from docker 就代表裝好了

貼上 terminal 之後按 enter

```
allow regular users to execute Docker

# Create the docker group.
sudo groupadd docker

# Add your user to the docker group
sudo usermod -aG docker $USER

newgrp docker

# verify
0 docker run hello-world
```

貼上 terminal 之後按 enter

Install Docker Compose

```
sudo apt-get update
sudo apt-get install docker-compose-plugin -y

# verify
docker compose version
```

```
Docker Compose version v2.35.1
```

以上是環境的安裝

以下開始建立語音的 code

Deploy Astra Application

1. Clone the code locally

```
1. cd -
2. git clone -b amazon-a11y https://github.com/chen188/Astra.a11y
```

貼到 terminal 後按 enter

2. Prepare configuration files

In the project's root directory, execute the following commands to provide configuration files.

```
1. cd ~/Astra.a11y
2.
3. # Create .env from the example
4. cp ../env.example ../env
5.
6. # Create property.json from the example
7. cp ../agents/property.json.example ../agents/property.json
8.
9. # Create .aws from example
10. cp ../playground/.env.example ../playground/.env
```

貼到 terminal 後按 enter

這部分不用貼

After that, you need to modify the .env file in the project code root directory, adjusting the following configurations. Others not listed can use default values:

```
1. # Agora App ID and Agora App Certificate
2. # required: this variable must be set
3. AGORA_APP_ID=your-ago-app-id
4. AGORA_APP_CERTIFICATE=your-ago-app-certificate
5.
6. # Extension: bedrock_llm
7. # Extension: polly_tts
8. # Extension: transcribe_asr
9. AWS_ACCESS_KEY_ID=your-aws-access-key-id
10. AWS_SECRET_ACCESS_KEY=your-aws-access-key
11.
12. # model id supported by Bedrock Converse API
13. AWS_BEDROCK_MODEL=us.amazon.nova-pro-v1.0
14.
15. AWS_REGION=us-west-2 # the Region you're using
```

接下來要進入 terminal 改環境變數

- Terminal 打 `vim .env` 按 enter 會看到藍色字體的 terminal
- 按 `i` 會出現 insert 這時候就可以開始改了，改完要離開要按 `i` 之後按 `esc`，之後再打一個 `:wq`
- 要改

```

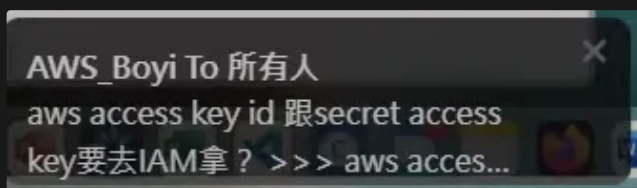
1  # Agora App ID and Agora App Certificate
2  # required: this variable must be set
3  AGORA_APP_ID=<your-ogora-app-id>
4  AGORA_APP_CERTIFICATE=<your-ogora-app-certificate>
5
6  # Extension: bedrock_llm
7  # Extension: polly_tts
8  # Extension: transcribe_asr
9  AWS_ACCESS_KEY_ID=<your-aws-access-key-id>
10 AWS_SECRET_ACCESS_KEY=<your-aws-access-key>
11
12 # model id supported by Bedrock Converse API
13 AWS_BEDROCK_MODEL=us.amazon.nova-pro-v1:0
14
15 AWS_REGION=us-west-2 # the Region you're using

```

agora_app_id、agora_app_certification、AWS_access_key_id、AWS_secret_access_key

她影片裏面沒有說這是從哪裡來的

(我不知道這些實際資訊會在哪裡，似乎會在文件裡面)



```

1  # Agora App ID and Agora App Certificate
2  # required: this variable must be set
3  AGORA_APP_ID=<your-ogora-app-id>
4  AGORA_APP_CERTIFICATE=<your-ogora-app-certificate>
5
6  # Extension: bedrock_llm
7  # Extension: polly_tts
8  # Extension: transcribe_asr
9  AWS_ACCESS_KEY_ID=<your-aws-access-key-id>
10 AWS_SECRET_ACCESS_KEY=<your-aws-access-key>
11
12 # model id supported by Bedrock Converse API
13 AWS_BEDROCK_MODEL=us.amazon.nova-pro-v1:0
14
15 AWS_REGION=us-west-2 # the Region you're using

```

```
AWS_REGION=us-west-2
```

- AWS_REGION 都是 us-west-2
- 要存好設定要按 esc 再打 :wq 這樣就好了

這樣就改完第一個 .env

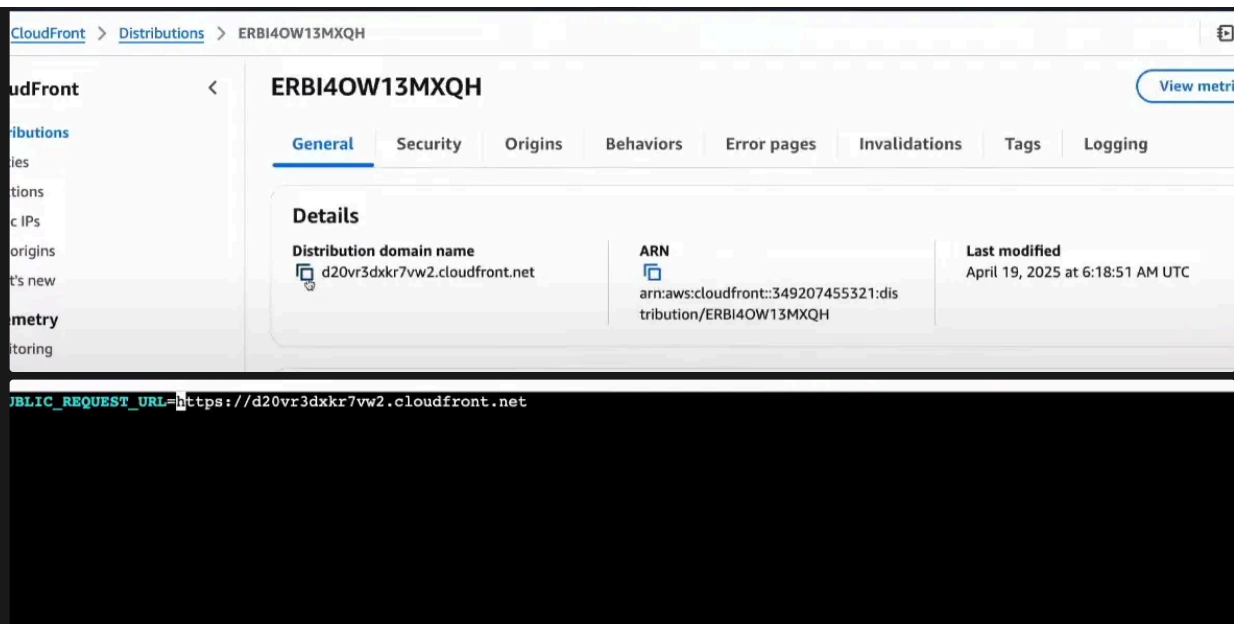
再來改第二個

```

buntu@ip-172-31-22-118:~/Astra.ai$ cd playground/
buntu@ip-172-31-22-118:~/Astra.ai/playground$ vim .env

```

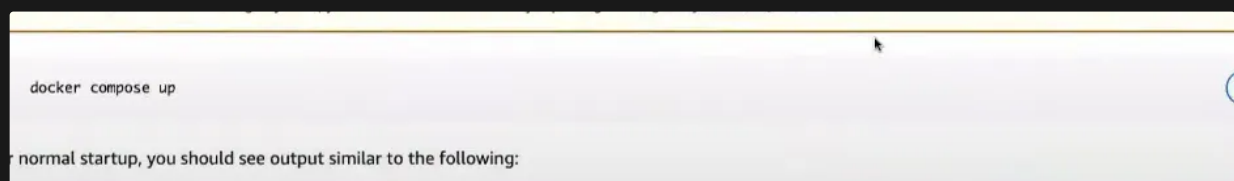
- 打 cd playground/ 進入後打 vim .env
- 回到 distribution 複製 8080 的 domain name



- esc : wq

以上就改好兩個環境變數

wq 回到上一層之後到文件複製



記得不要在 playground 執行，要打 cd.. 回到 Astra.ai

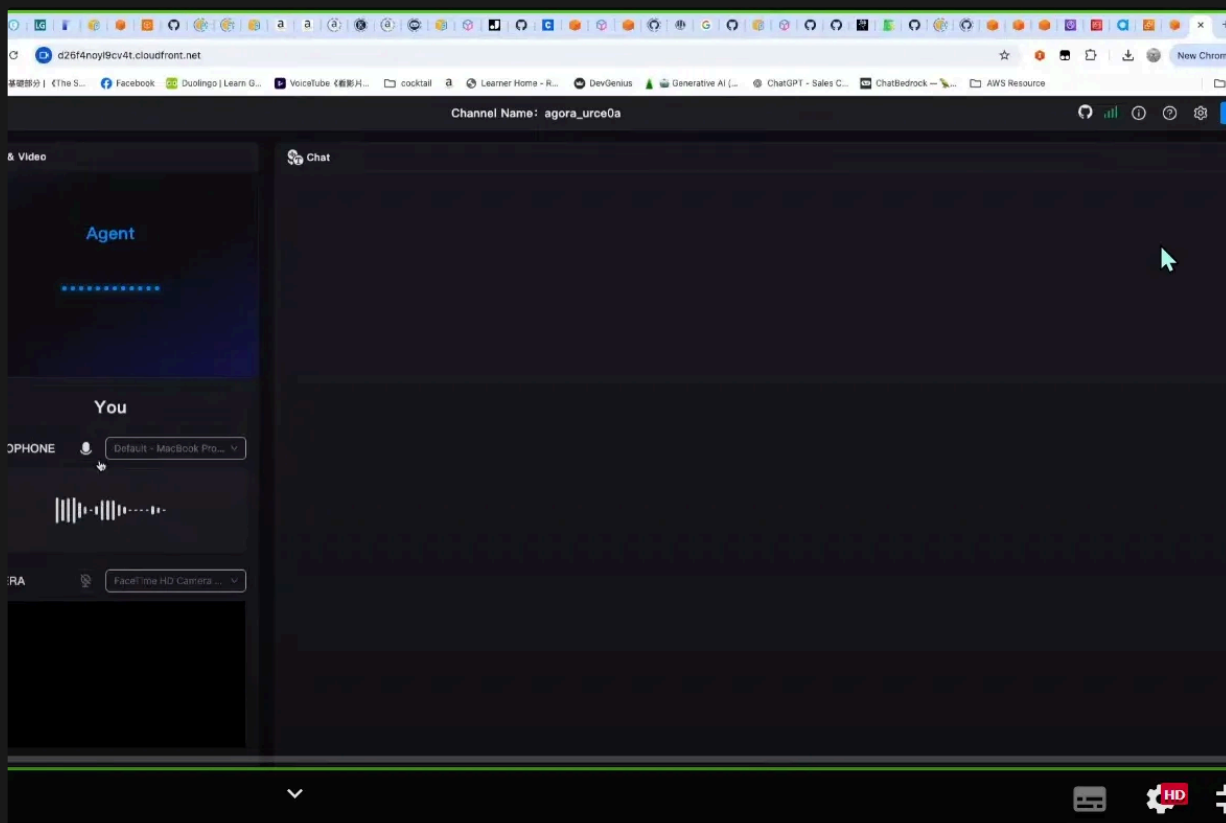
- 貼上 docker compose up



出現 local host 3000 就完成了

```
21 Astra_playground_dev | > Astra-playground@0.1.0 start
22 Astra_playground_dev | > next start
23 Astra_playground_dev |
24 Astra_playground_dev | ▲ Next.js 14.2.4
25 Astra_playground_dev | - Local: http://localhost:3000
26 ...
```

3000 的 port 的 domain name 也先複製起來，直接貼在網址搜尋就可以進入語音助理的前端



Settings

General Prompt MCP

GRAPH NAME

① [Chat Mode] Transcribe ASR -> Bedrock LLM -> Polly TTS

Bedrock Base - Chat

AGENT MODE

Chat

INPUT LANGUAGE

Chinese Traditional

VOICE

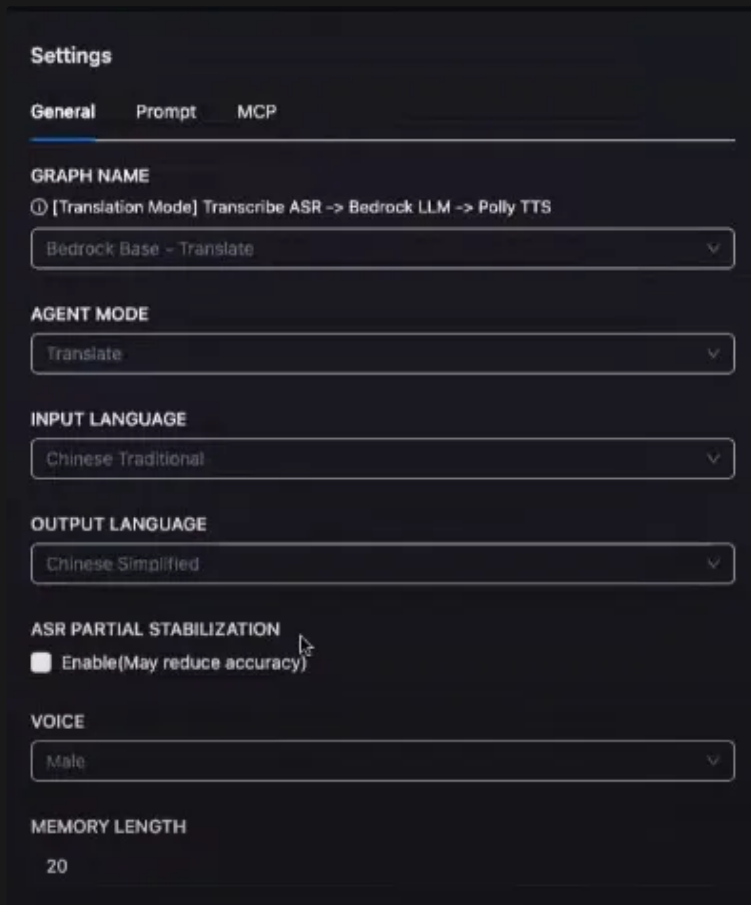
Male

MEMORY LENGTH

20

設定完之後案右上角的 connect

如果是粵語的話就再進到設定 選 bedrock base - translate , 再改 output 為 chinese Simplified 改完後再切回 bedrock base chat , input 也要是 chinese traditional



Settings

General Prompt MCP

GRAPH NAME
① [Translation Mode] Transcribe ASR -> Bedrock LLM -> Polly TTS
Bedrock Base - Translate

AGENT MODE
Translate

INPUT LANGUAGE
Chinese Traditional

OUTPUT LANGUAGE
Chinese Simplified

ASR PARTIAL STABILIZATION
☒ Enable(May reduce accuracy)

VOICE
Male

MEMORY LENGTH
20

這樣語音前端就完全建立完成了

CloudFront (要建立兩個 CloudFront)

1. 搜尋 CloudFront (紫色的圖)

- create CloudFront distribuion
- Origin domain - 回到剛剛建立的 instance - 複製右手邊的 IPv4 DNS

Public IPv4 DNS

 ec2-54-244-192-175.us-west-2.compute.amazonaws.com
| open address 

- 回到 CloudFront origin

Origin

Origin domain
Choose an AWS origin, or enter your origin's domain name. Learn more [↗](#)

Enter a valid DNS domain name, such as an S3 bucket, HTTP server, or VPC origin ID.

Protocol [info](#)

☐ HTTP only

☒ HTTPS only

☐ Match viewer

HTTPS port
Enter your origin's HTTPS port. The default is port 443.

Minimum Origin SSL protocol
The minimum SSL protocol that CloudFront uses with the origin.

☒ TLSv1.2

☐ TLSv1.1

☐ TLSv1

☐ SSLv3

Default cache behavior

Path pattern [info](#)

Compress objects automatically [info](#)

☐ No

☒ Yes

Viewer

Viewer protocol policy

☐ HTTP and HTTPS

☒ Redirect HTTP to HTTPS

☐ HTTPS only

Allowed HTTP methods

☐ GET, HEAD

☐ GET, HEAD, OPTIONS

☒ GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE

Cache HTTP methods
GET and HEAD methods are cached by default.

☐ OPTIONS

Allow gRPC requests over HTTP/2 [info](#)

☐ Enable

Restrict viewer access
If you restrict viewer access, viewers must use CloudFront signed URLs or signed cookies to access your content.

☒ No

☐ Yes

- Create distribution
- Firewall 選 Do not enable
- create - 這樣建立完第一個

回到上一動 - 建立下一個

Distributions [info](#)

[Option+S]

	Status	Description	Type	Domain name	Alternate domain...	Origins
Loading distributions						

- 右上角 create
- 接下來動作都跟上一部一樣，只有 HTTPs port 改 3000

Default cache behavior

Path pattern [Info](#)

Default (*)

Compress objects automatically [Info](#)

- ☐ No
- ☒ Yes

Viewer

Viewer protocol policy

- ☐ HTTP and HTTPS
- ☒ Redirect HTTP to HTTPS
- ☐ HTTPS only

Allowed HTTP methods

- ☐ GET, HEAD
- ☐ GET, HEAD, OPTIONS
- ☒ GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE

Cache HTTP methods

GET and HEAD methods are cached by default.

- ☐ OPTIONS