

App Dev Pills



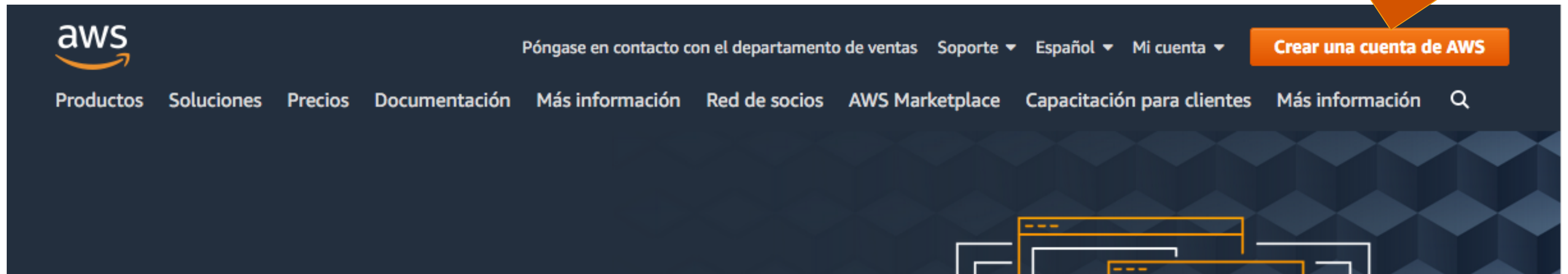
Despliegue de Flask en AWS

Intro a AWS y algo de linux



■ Alta en AWS

- Entramos en la dirección de Amazon Web Services
 - <https://aws.amazon.com/es>
- Seleccionamos botón Crear una cuenta de AWS



■ Alta en AWS

- Rellenar los datos de la cuenta
- Continuar

Crear una cuenta de AWS

Dirección de correo electrónico

Contraseña

Confirmar contraseña

Nombre de la cuenta de AWS ⓘ

Continuar

[Iniciar sesión en una cuenta de AWS existente](#)

© 2020 Amazon Web Services, Inc. o sus empresas afiliadas.
Todos los derechos reservados.
[Política de privacidad](#) | [Términos de uso](#)



Alta en AWS

- Rellenar los datos de contacto
 - Tipo de cuenta: Personal
 - País/Región: España
 - Resto de datos. Los tuyos

Información de contacto

Todos los campos son obligatorios.

Seleccione el tipo de cuenta y escriba sus datos de contacto en los campos a continuación.

Tipo de cuenta ⓘ
☐ Profesional ☒ Personal

Nombre completo

Número de teléfono

País/Región

Dirección

Ciudad

Estado, provincia o región

Código postal

☐ Seleccione aquí para indicar que ha leído y está conforme con los términos del [Contrato de cliente de AWS](#)

Crear cuenta y continuar

© 2020 Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados.



Alta en AWS

- Rellenar los datos de pago

Datos reales o no podrás abrirla.


No realizan cobros tal y como indican.

Al año, dar de baja todos los servicios para evitar cobros



Información de pago

Usamos su información de pago para verificar su identidad y solo en caso de que exceda [los límites de capa gratuita de AWS](#). No le cobraremos por el uso que haga por debajo de los límites de capa gratuita de AWS. Para obtener más información, consulte [las preguntas más frecuentes](#).

 Cuando envíe su información de pago, le haremos un cargo de 1 USD/EUR en su tarjeta de crédito como cargo de verificación para comprobar que su tarjeta es válida. La cantidad se mostrará como pendiente en el extracto de su tarjeta de crédito durante un periodo de 3 a 5 días hasta que se complete la verificación; entonces, el cargo se eliminará. Se le redirigirá al sitio web de su banco para autorizar el cargo de verificación.

Número de tarjeta de crédito o débito

Fecha de vencimiento

Nombre del titular de la tarjeta

Dirección de facturación

☒ Utilizar mi dirección de contacto

Miguel Servet, 10, 16
Valencia Valencia 46015
ES

☐ Utilizar una nueva dirección

Verificar y añadir

© 2020 Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados.

[Política de privacidad](#) | [Términos de uso](#) | [Cerrar sesión](#)

Alta en AWS

- Confirmar identidad

La llamada de voz puede ser en inglés

Yo prefiero SMS

Confirme su identidad

Para poder usar su cuenta de AWS, debe verificar su número de teléfono. Cuando continúe, el sistema automatizado de AWS se pondrá en contacto con usted para facilitarle un código de verificación.

¿Cómo deberíamos enviarle el código de verificación?

- ☒ Mensaje de texto (SMS)
☐ Llamada de voz

Código de país o región

España (+34)

Número de teléfono móvil

Comprobación de seguridad



Escriba los caracteres que se muestran arriba

Enviar SMS

© 2020 Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados.

[Política de privacidad](#) | [Términos de uso](#) | [Cerrar sesión](#)



Alta en AWS

- Confirmar identidad

La llamada de voz puede ser en inglés

Yo prefiero SMS

Introducir código SMS



Mensaje de texto (SMS)


Escriba el código de verificación

Escriba el código de verificación de cuatro dígitos que recibió en su teléfono.

Verificar código

¿Tiene algún problema? A veces se tarda hasta 10 minutos en recibir el código de verificación. Si ha transcurrido más tiempo, [vuelva a la página anterior](#) y escriba de nuevo el número.

Mensaje de texto (SMS)



Su identidad se ha verificado correctamente.




Continuar

Alta en AWS

- Elegir plan
Gratis, lógicamente

Seleccione un plan

AWS ofrece una gran variedad de planes para satisfacer sus necesidades. Elija el que mejor se ajuste al uso que haga de AWS. [Más información](#)

 Plan Basic	 Plan Developer	 Plan Business
Gratis	Desde 29 USD al mes	Desde 100 USD al mes
<ul style="list-style-type: none">• Incluido con todas las cuentas• Acceso automático ininterrumpido a foros y recursos• Comprobaciones de prácticas recomendadas para mejorar la seguridad y el desempeño• Acceso al estado y a las notificaciones	<ul style="list-style-type: none">• Para la adopción temprana, las pruebas y el desarrollo• Acceso por correo electrónico a AWS Support en horario laboral• El contacto principal puede abrir un número ilimitado de incidencias de soporte• Tiempos de respuesta de 12 horas para sistemas que no son de producción	<ul style="list-style-type: none">• Orientado a cargas de trabajo de producción y dependencias críticas para la empresa• Acceso ininterrumpido a AWS Support por chat, teléfono y correo electrónico• Cualquier usuario puede abrir un número ilimitado de incidencias de soporte• Tiempos de respuesta de 1 hora para sistemas de producción

¿Necesita soporte de nivel Enterprise?

Póngase en contacto con su responsable de cuenta para obtener más información acerca de la ejecución de cargas de trabajo críticas y empresariales en AWS (desde 15 000 USD al mes). [Más información](#)

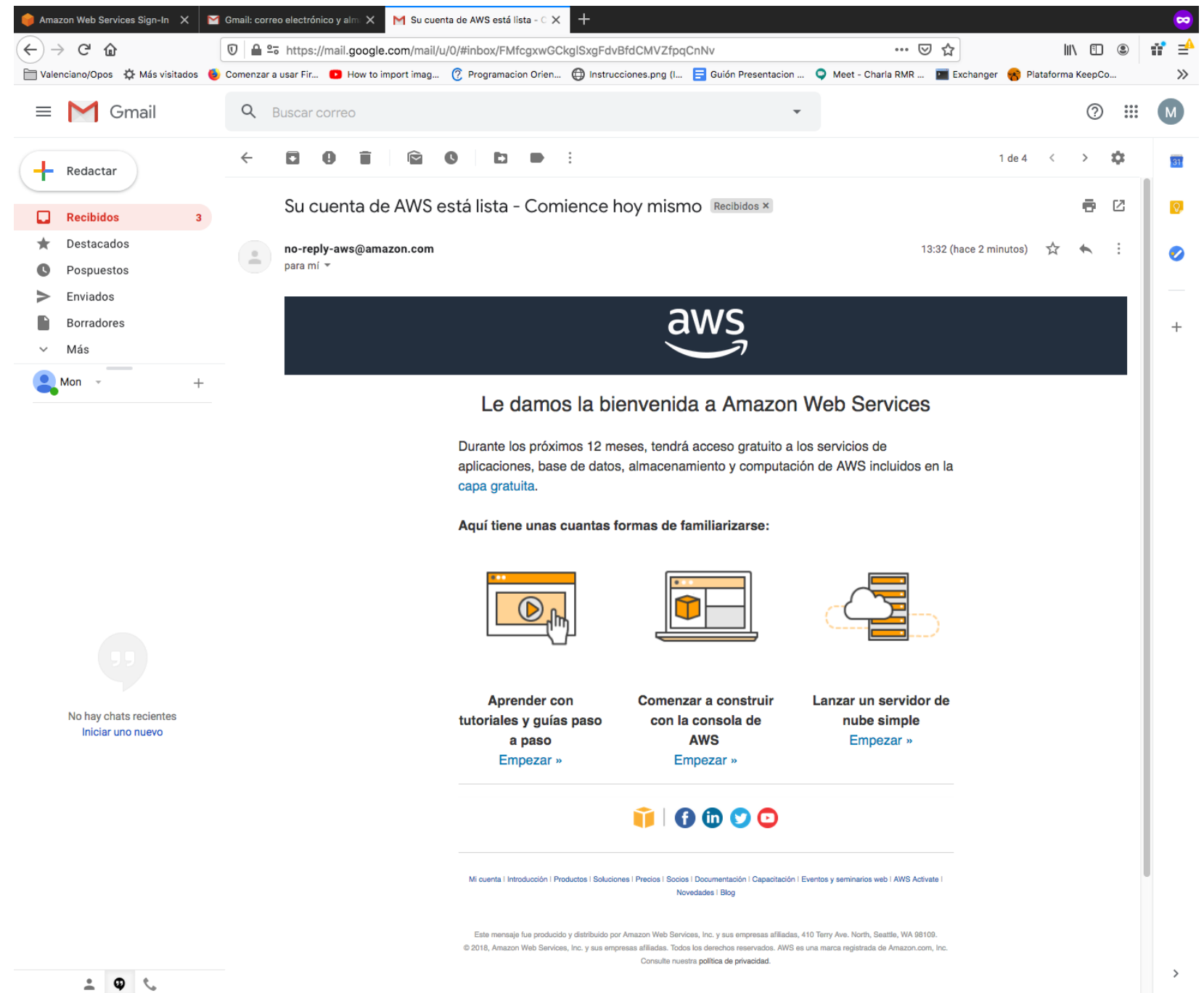
© 2020 Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados.

[Política de privacidad](#) | [Términos de uso](#) | [Cerrar sesión](#)



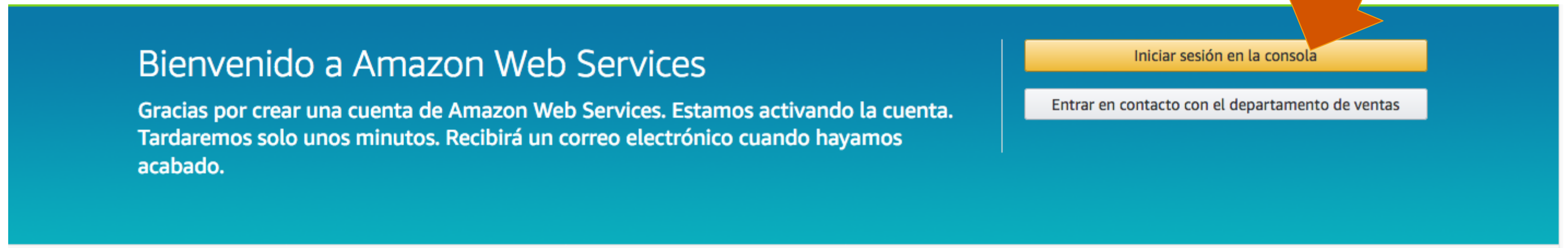
Alta en AWS

- Hecho
 - Ir al correo para verificar alta correcta



■ Alta en AWS

- Hecho
 - Entrar en la consola



■ Alta en AWS

- Rellenar los datos del formulario
 - Tipo de cuenta: Personal
 - País/Región: España
 - Resto de datos. Los tuyos



■ Alta en AWS

- Utilizaremos durante un año la capa gratuita
 - **EC2 (Elastic computing)**

Crear Instancia con Ubuntu, BBDD (sqlite), servidor Web, ... y python
750 h/mensuales (un mes tiene 744)



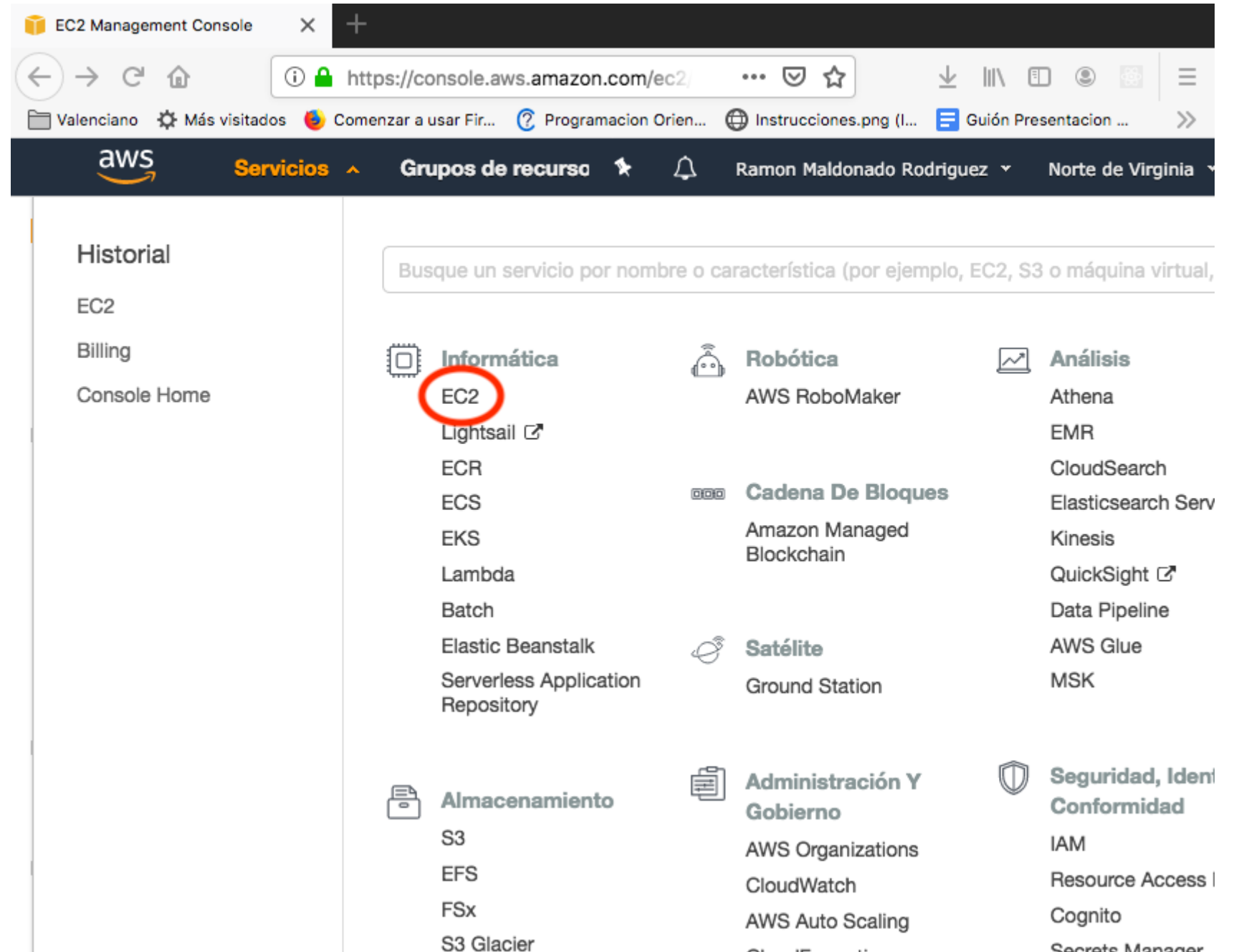
■ Alta en AWS

- Necesitamos
 - Una tarjeta de crédito
 - Una dirección de correo
 - Activar el alta



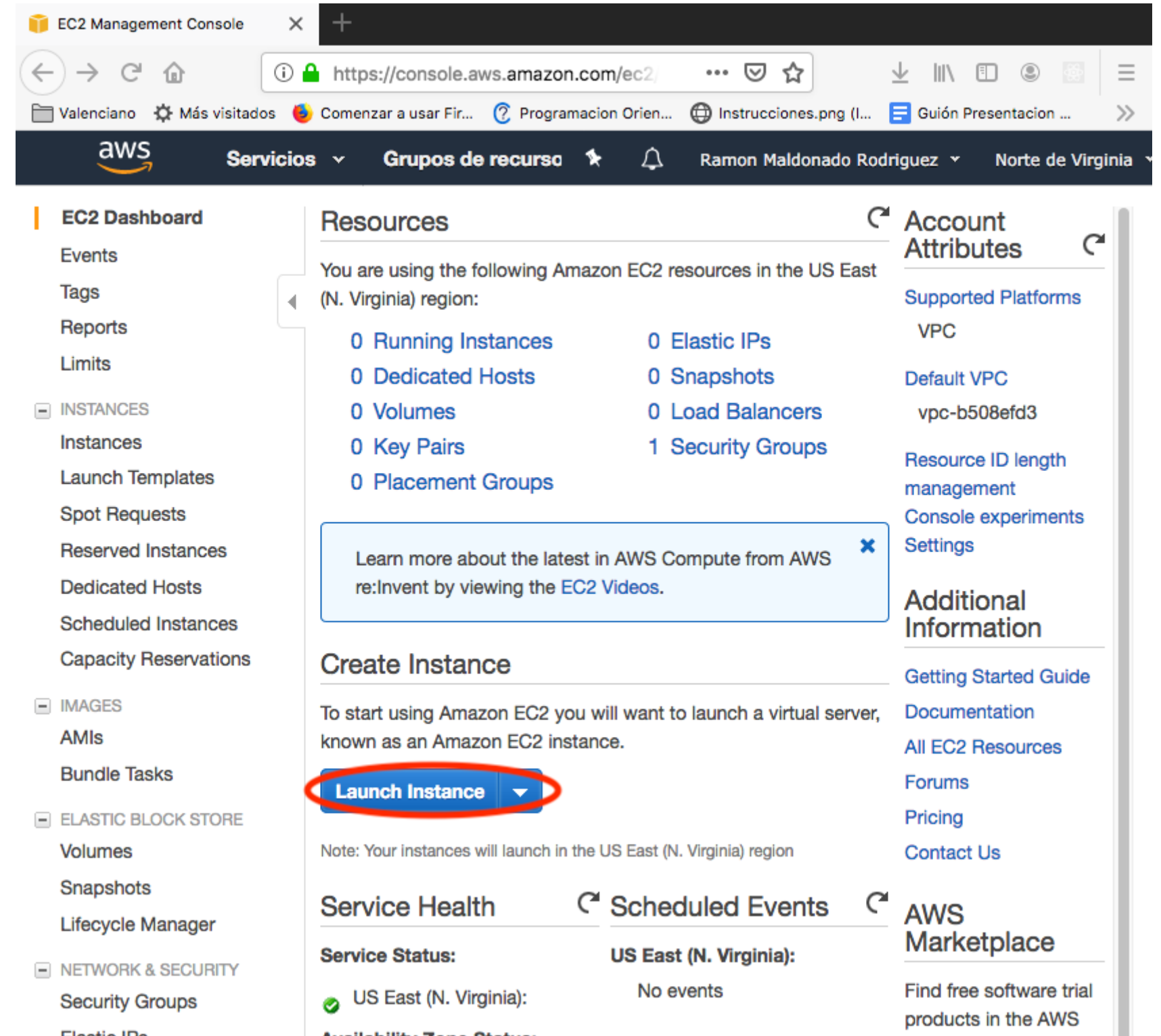
■ Nuestra instancia EC2

- Buscar EC2



■ Nuestra instancia EC2

- Crear instancia

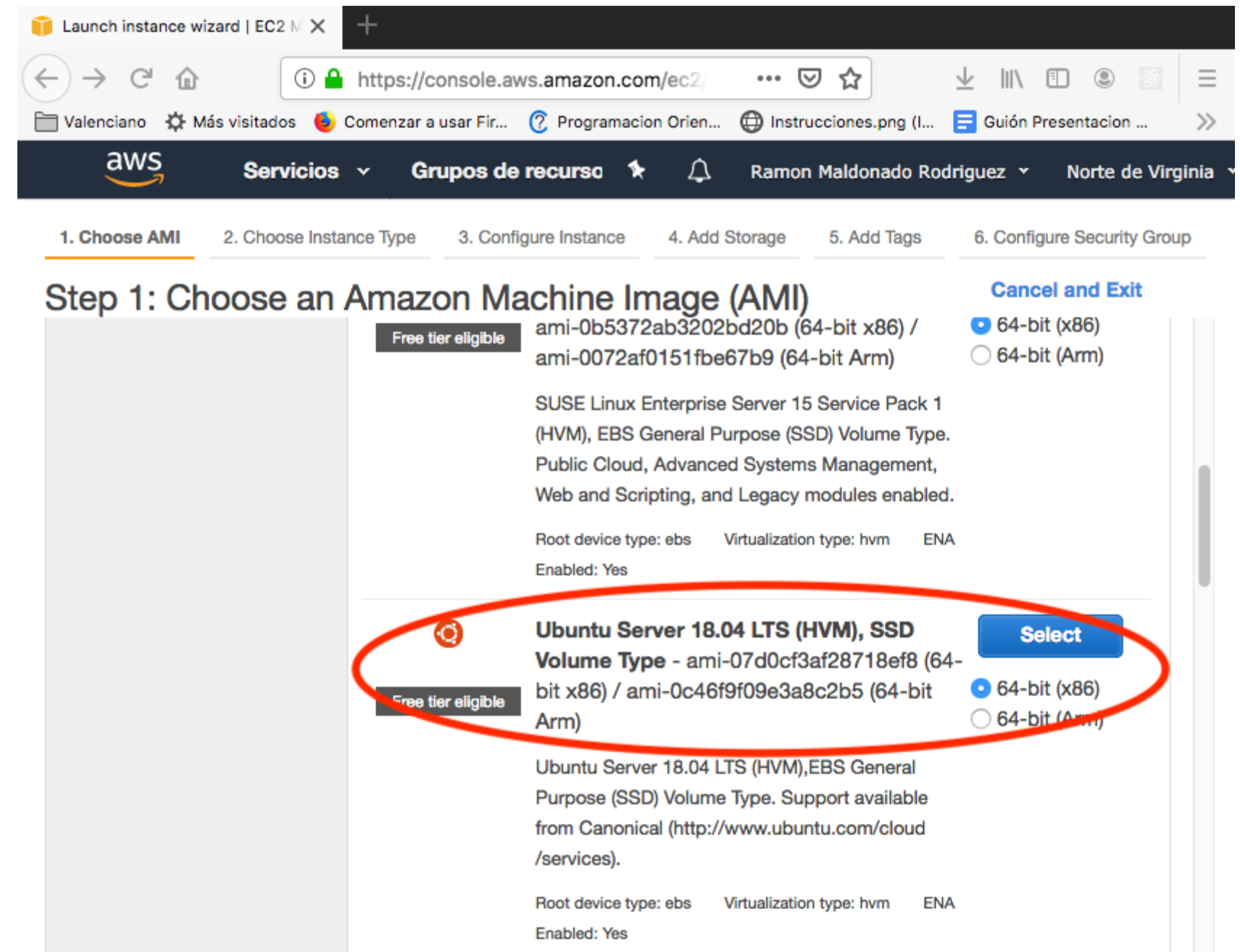


The screenshot displays the AWS Management Console for the EC2 service. The left-hand navigation pane shows the 'EC2 Dashboard' with various options like Events, Tags, Reports, Limits, INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The main content area is titled 'Resources' and shows a summary of EC2 resources in the US East (N. Virginia) region, including 0 Running Instances, 0 Elastic IPs, 0 Dedicated Hosts, 0 Snapshots, 0 Volumes, 0 Load Balancers, 0 Key Pairs, 1 Security Groups, and 0 Placement Groups. Below this, a 'Create Instance' section provides instructions on launching a virtual server. A blue button labeled 'Launch Instance' is prominently displayed and circled in red. To the right of the main content, there are sections for 'Account Attributes' and 'Additional Information', including links to the 'Getting Started Guide', 'Documentation', 'All EC2 Resources', 'Forums', 'Pricing', and 'Contact Us'.



■ Nuestra instancia EC2

- Seleccionar Sistema Operativo
 - Ubuntu



■ Nuestra instancia EC2

- Seleccionar Tamaño

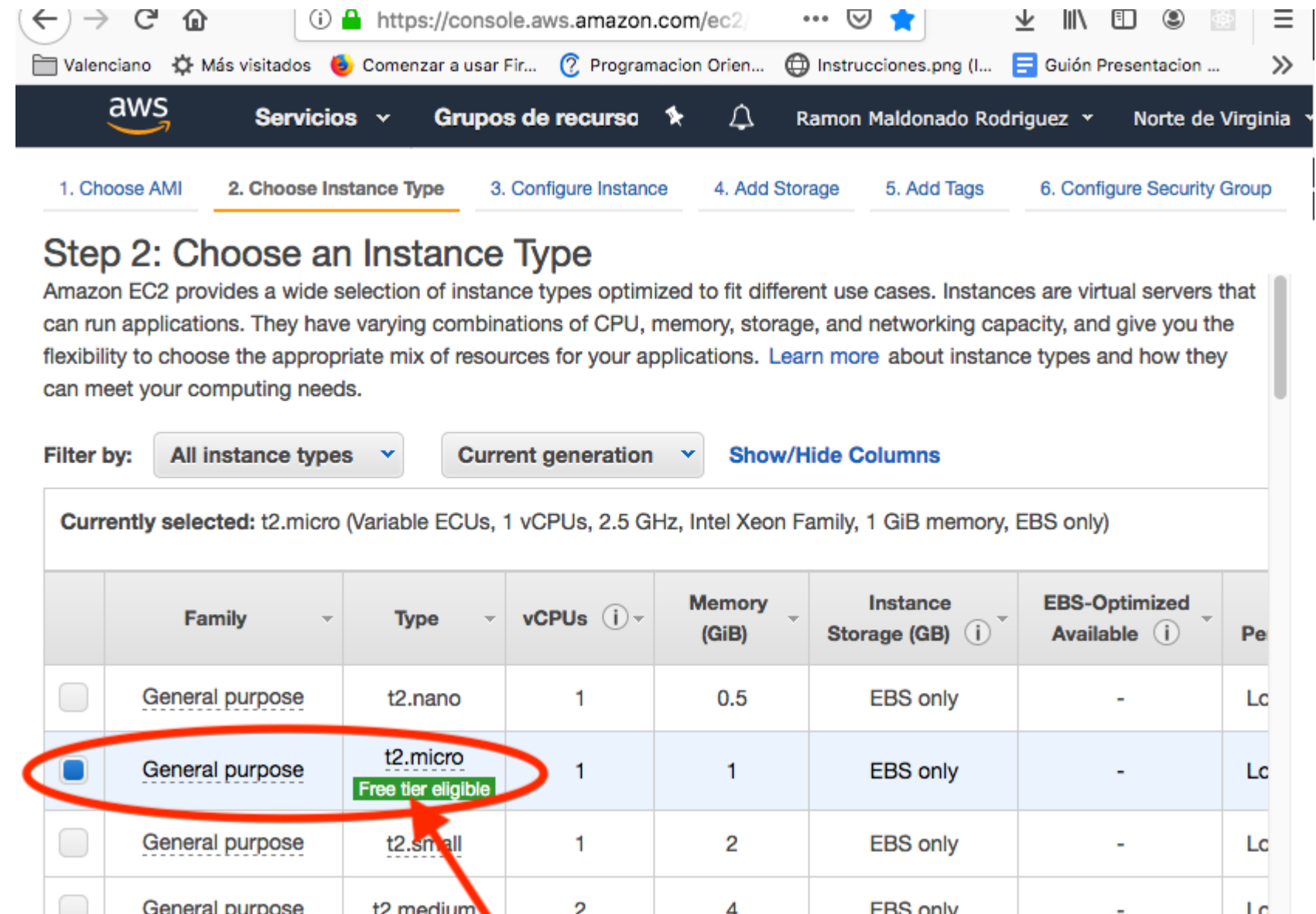
- t2.micro

- Next

Next: Configure Instance Details

- No

~~Review & Launch~~



aws Servicios Grupos de recurso Ramon Maldonado Rodriguez Norte de Virginia

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Per
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Lc
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Lc
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Lc
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Lc

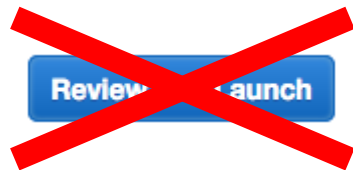


■ Nuestra instancia EC2

- Detalles de instancia
 - Se deja como está

Next: Add Storage

- No



Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: **All instance types** **Current generation** [Show/Hide Columns](#)

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Price
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low



■ Nuestra instancia EC2

- Almacenamiento
 - Tomamos el máximo permitido
 - 30 GB
- No



https://console.aws.amazon.com/ec2/

Valenciano Más visitados Comenzar a usar Fir... Programacion Orien... Instrucciones.png (I... Guión Presentacion ...

aws Servicios Grupos de recurso Ramon Maldonado Rodriguez Norte de Virginia

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Per
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Lc
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Lc
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Lc
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Lc



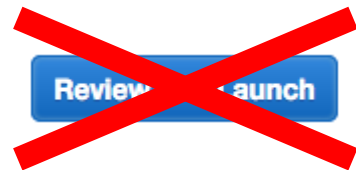
■ Nuestra instancia EC2

- Tags

- Continuar

Next: Configure Security Group

- No



The screenshot shows the AWS Management Console interface during the 'Launch instance wizard'. The browser tabs include 'Launch instance wizard', 'Políticas de IAM para Am...', 'Capa gratuita de AWS', and 'que es iam aws - Buscar'. The address bar shows 'https://console.aws.amazon.com/'. The navigation bar includes the AWS logo, 'Servicios', 'Grupos de recurso', and user information 'Ramon Maldonado Rodriguez'. The wizard progress bar shows five steps: '1. Choose AMI', '2. Choose Instance Type', '3. Configure Instance', '4. Add Storage', and '5. Add Tags' (which is the active step).

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	Value	Instances	Volumes
Name	flask BZ	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

[Add another tag](#) (Up to 50 tags maximum)

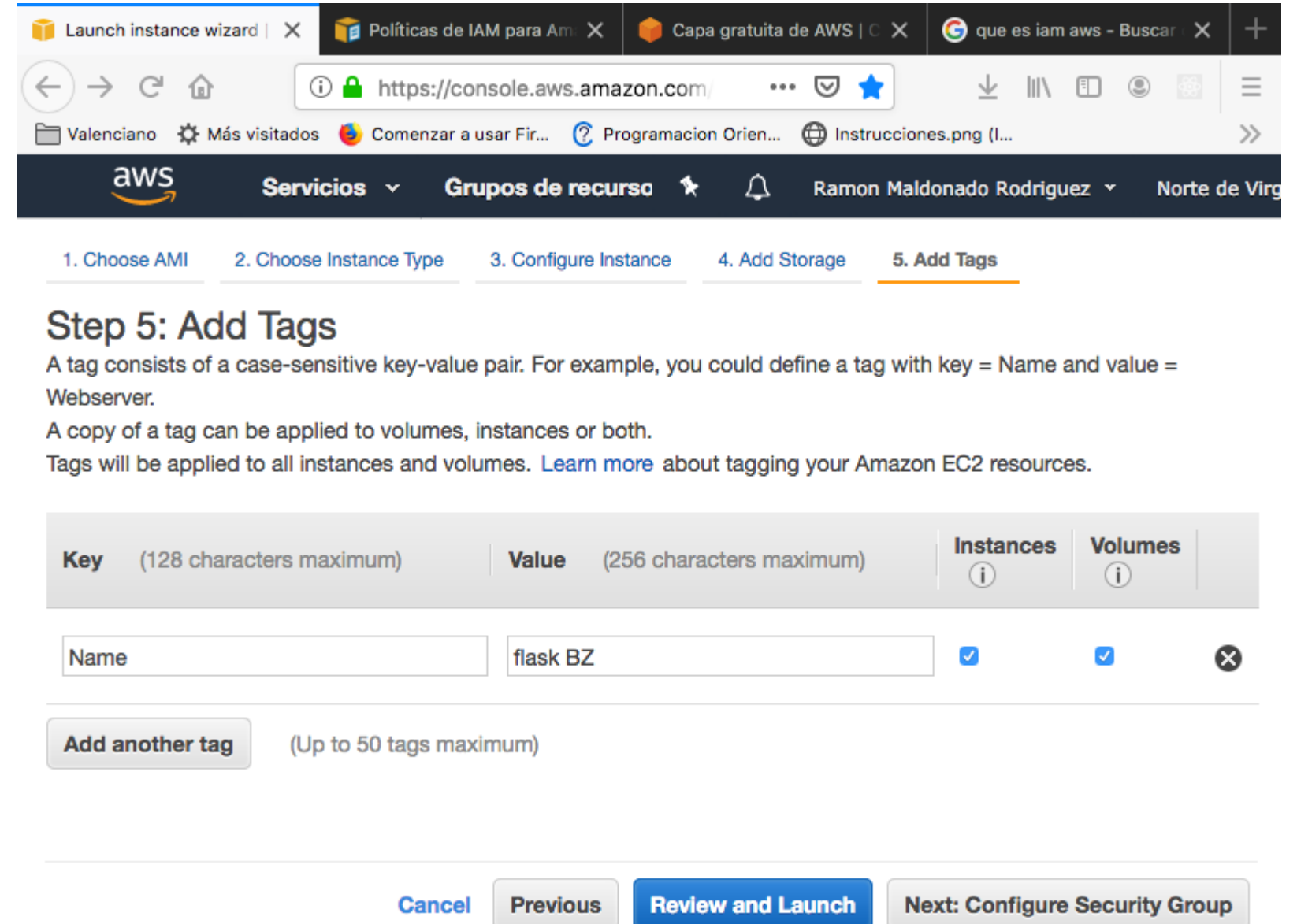
At the bottom, there are four buttons: 'Cancel', 'Previous', 'Review and Launch' (highlighted in blue), and 'Next: Configure Security Group'.



■ Nuestra instancia EC2

- Security Group
 - Acceso a puerto 22
 - Conexión SSH
- Por fin

Review and Launch



The screenshot shows the AWS Management Console interface during the 'Launch instance wizard'. The browser tabs include 'Launch instance wizard', 'Políticas de IAM para Am...', 'Capa gratuita de AWS', and 'que es iam aws - Buscar'. The address bar shows 'https://console.aws.amazon.com/'. The navigation bar includes the AWS logo, 'Servicios', 'Grupos de recurso', and the user name 'Ramon Maldonado Rodriguez'. The wizard progress bar shows five steps: '1. Choose AMI', '2. Choose Instance Type', '3. Configure Instance', '4. Add Storage', and '5. Add Tags' (which is the active step).

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	Value	Instances	Volumes
Name	flask BZ	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

[Add another tag](#) (Up to 50 tags maximum)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Security Group](#)



Nuestra instancia EC2

- Lanzarlo
 - Nos permite revisar y modificar

Review and Launch



Launch instance wizard | Políticas de IAM para Am... | Capa gratuita de AWS | que es iam aws - Buscar

https://console.aws.amazon.com/

aws Servicios Grupos de recurso Ramon Maldonado Rodriguez Norte de Virg

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

⚠ Improve your instances' security. Your security group, launch-wizard-1, is open to the world.

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

▼ AMI Details [Edit AMI](#)

Free tier eligible **Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-07d0cf3af28718ef8**

Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Root Device Type: ebs Virtualization type: hvm

▼ Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

▼ Security Groups [Edit security groups](#)

Security group name launch-wizard-1

Description launch-wizard-1 created 2019-07-29T16:23:27.316+02:00

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ
SSH	TCP	22	0.0.0.0/0	

► Instance Details [Edit instance details](#)

► Storage [Edit storage](#)

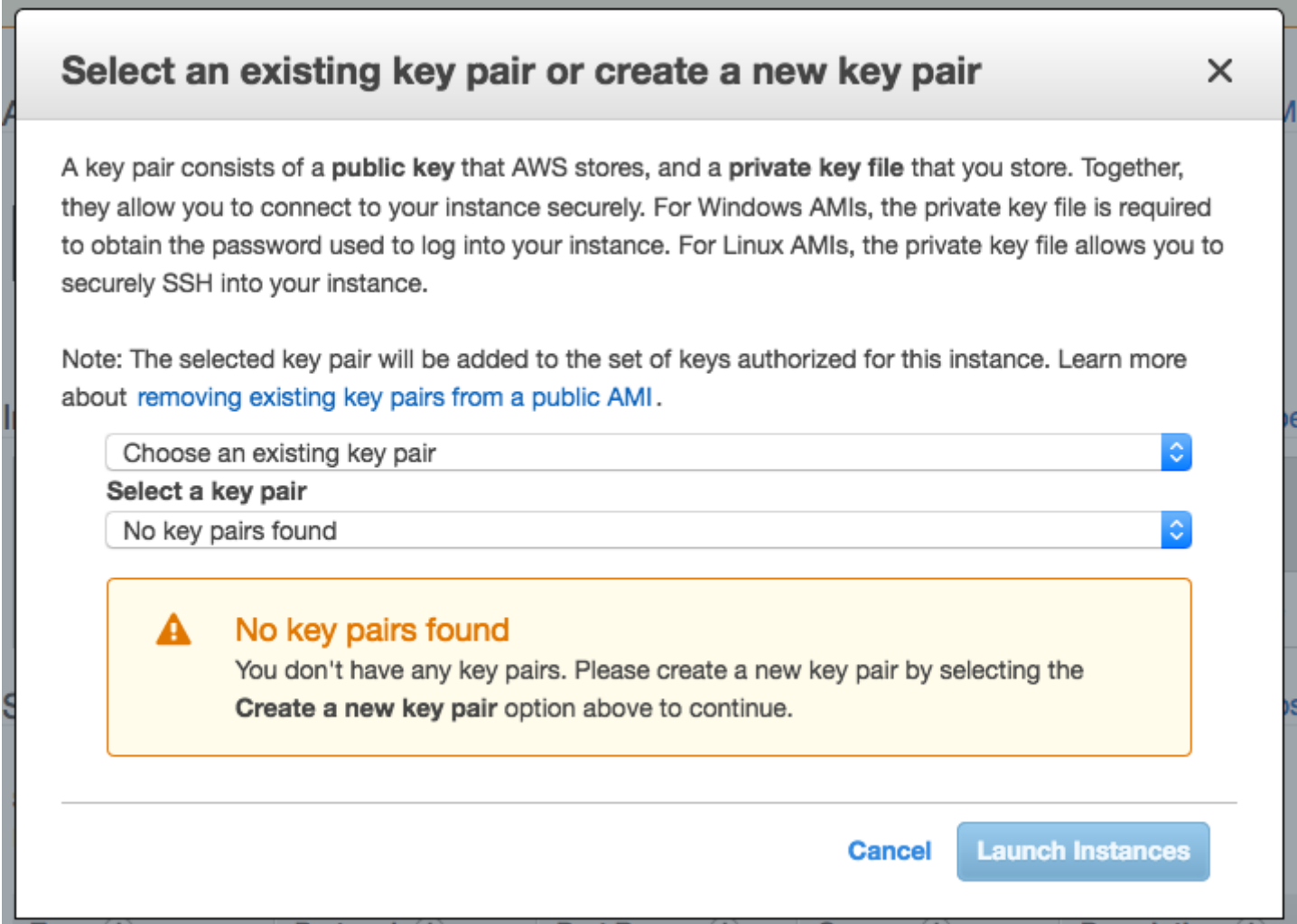
[Cancel](#) [Previous](#) [Launch](#)

Comentarios Español Política de privacidad Términos de uso

© 2008 - 2019, Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados.

■ Nuestra instancia EC2

- El acceso SSH:22 se hace por
 - par clave pública/privada
- Debe crearse
 - guardarse con mucho cuidado o no se podrá acceder
- Descargar
 - cambiar permiso a 400 (mac/linux)
 - `chmod 400 clave.pem`
- Lanzar



Select an existing key pair or create a new key pair ✕

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair

Select a key pair

No key pairs found

⚠ No key pairs found

You don't have any key pairs. Please create a new key pair by selecting the **Create a new key pair** option above to continue.

[Cancel](#) [Launch Instances](#)



■ Acceso EC2 desde el terminal

- Abrimos terminal y escribimos

```
ssh -i nombre.pem ubuntu@ip_aws
```

```
1. ubuntu@ip-172-31-63-64: ~ (ssh)
kc@MacBook-Pro-de-kc:~/Downloads > chmod 400 bootZ3.pem
kc@MacBook-Pro-de-kc:~/Downloads > ssh -i bootZ3.pem ubuntu@3.91.222.101
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-1044-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

System information as of Tue Jul 30 11:50:10 UTC 2019

System load:  0.0               Processes:            91
Usage of /:   3.6% of 29.02GB   Users logged in:     1
Memory usage: 15%              IP address for eth0: 172.31.63.64
Swap usage:   0%

0 packages can be updated.
0 updates are security updates.

Last login: Tue Jul 30 11:41:50 2019 from 139.47.116.131
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-63-64:~$
```



■ Pasos a realizar en instancia

- Instalación de todo lo necesario
 - nginx: Servidor web
 - git: gestión de versiones
 - circus: gestor de procesos
 - dependencias de python
 - make, build-essential, zlib1g-dev, libssl-dev, libbz2-dev
 - libreadline-dev, libsqlite3-dev, libffi-dev, gunicorn
 - Otros usos
 - curl, sqlite3

```
sudo apt-get install -y nginx git circus build-essential zlib1g-dev libssl-dev libbz2-dev libreadline-dev libsqlite3-dev libffi-dev gunicorn curl sqlite3
```



- Creación de usuario de la app
 - `sudo adduser <appname>`
 - `sudo passwd -l <appname>`
- Añadir nginx-user al grupo appname
 - `sudo adduser www-data <appname>`





- Crear el entorno virtual

- Como usuario <appname>

- `sudo -u <appname> -i`

- Instalar pyenv

- `git clone https://github.com/yyuu/pyenv.git ~/.pyenv`

- `git clone https://github.com/yyuu/pyenv-virtualenv.git ~/.pyenv/
plugins/pyenv-virtualenv`



- Crear el entorno virtual (2)

- configurar pyenv

- `echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.bash_profile`
- `echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.bash_profile`
- `echo 'eval "$(pyenv init -)"' >> ~/.bash_profile`
- `echo 'eval "$(pyenv virtualenv-init -)"' >> ~/.bash_profile`

- ejecutar configuración pyenv

- `source ~/.bash_profile`



- Crear el entorno virtual (y 3)
 - instalar la versión de python que queremos
 - `pyenv install 3.7.4`
 - crear el entorno virtual con esa versión de python
 - `pyenv virtualenv 3.7.4 <nombre_de_entorno_virtual>`



- Instalar nuestra app en flask
 - clonar la última versión estable de nuestro repo
 - `git clone https://github.com/...`
 - crear carpeta de logs. (necesarios para app, circus y nginx)
 - `mkdir logs`



- Configurar nuestra app flask
 - Copiar config_template.py > config.py
 - `cp <appname>/config_template.py <appname>/config.py`
 - Editar config_template.py
 - `nano <appname>/config.py`
 - Añadir SECRET_KEY=<nuevo_valor>



- Instalar dependencias de nuestra app
 - Activar entorno virtual
 - `pyenv activate <nombre_entorno_virtual>`
 - Upgrade de pip
 - `pip install --upgrade pip`
 - Instalar dependencias
 - `pip install -r <appname>/requirements.txt`



- Instalar WSGI

- WSGI: Web Server Gateway Interface (Interfaz de puerta de enlace del servidor web)

- `pip install gunicorn`



- Probar nuestra app
 - Exportar variables de entorno
 - `export FLASK_APP=run.py`
 - Lanzar app
 - `cd <appname>`
 - `gunicorn run:app`
 - Comprobar que funciona
 - `curl http://localhost:8000/`





Pasos a realizar en instancia

- Presentar nuestra app en internet
 - Abrir puerto 80
 - Ir a EC2 > Security Groups
 - Seleccionar el de la instancia (launch-wizard-1) y crear regla
 - Seleccionar Pestaña inbound > Edit > Add rule

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ	
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop	✕
HTTP	TCP	80	Custom 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop	✕

Add Rule

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Cancel Save



- Presentar nuestra app en internet

- Salir de usuario <appname>

- logout

- Como ubuntu, crear configuracion

- `sudo nano /etc/circus/conf.d/<appname>.ini`

Incluir el contenido

- relanzar circus (con nueva conf)

- `sudo service circusd restart`

- `sudo service circusd status`

```
[watcher: appname]  
working_dir = /home/username/appname  
cmd = gunicorn  
args = -w 1 -t 180 --pythonpath=. -b 0.0.0.0:8000 run:app  
uid = username  
numprocesses = 1  
autostart = true  
send_hup = true  
stdout_stream.class = FileStream  
stdout_stream.filename = /home/username/logs/  
gunicorn.stdout.log  
stdout_stream.max_bytes = 10485760  
stdout_stream.backup_count = 4  
stderr_stream.class = FileStream  
stderr_stream.filename = /home/username/logs/  
gunicorn.stderr.log  
stderr_stream.max_bytes = 10485760  
stderr_stream.backup_count = 4  
copy_env = true  
virtualenv = /home/username/.pyenv/versions/nombre_entorno_virt  
virtualenv_py_ver = 3.7  
[env: appname]
```



- ¿Dónde estamos?
 - `curl localhost:8000`
 - `<ip_aws>`
 - `<ip_aws:8000>`
- Debemos redirigir nuestra app al puerto 80





- Configurar nginx

- `sudo nano /etc/nginx/sites-available/<app_name>`

Incluir el contenido →

- Crear enlace simbolico en sites-enabled

- `sudo ln -s /etc/nginx/sites-available/<app_name> /etc/nginx/sites-enabled/<app_name>`

- Comprobar configuracion

- `sudo nginx -t`

- Reiniciar servicio

- `sudo service nginx restart`

```
server {
    listen 80;
    server_name <ip_aws>;
    access_log /home/username/logs/nginx-
access.log;
    error_log /home/username/logs/nginx-
error.log;
    root /home/username/appname/;
    client_max_body_size 10M;
    location /static {
        alias /home/username/appname/app/
static;
    }
    location / {
        include proxy_params;
        proxy_pass http://0.0.0.0:8000;
    }
}
```



- Localizar procesos activos por nombre

- `ps aux | grep <nombre>`

```
k5_flask 32656 0...
```



- Matar proceso por número

- `sudo kill -9 <process_number>`

