

Copy and Paste: Configuración automática de BGP con un café en la mano

Carlos Fraga



Índice ...

- Presentación de un amigo
- Hablando de cosas con un café
- Libros de cosas
- Resolviendo los problemas con café
- ¿Necesito más café ?
- Nuevas necesidades y problemas
- ¿ Loop infinito ?
- Recursos para seguir aprendiendo

Os presento a mi amigo:



Nombre: **BruGuP**

Edad: **179** (años terrestres)

Recién titulado en la escuela de su Sistema

Intereses tech:

- **TX** interplanetaria usando **taquiones**
 - **Networking cuántico**
- (si alguien tiene info que se ponga en contacto conmigo)

Aficiones Terrestres:

- **Café**

Problemas de nuestro amigo

- BruGuP tiene que trabajar para pagarse los cafés
- Trabaja en una empresa y su responsable le ha asignado la tarea de configurar BGP en unos dispositivos (Para la semana que viene)

¿Por dónde empiezo ? ---> TOMANDO UN CAFÉ !!!!! (en mi casa)

Características principales de BGPv4

Enrutamiento interdominio: BGP es un protocolo de enrutamiento utilizado para intercambiar información de enrutamiento entre diferentes dominios de sistemas autónomos.

Enrutamiento basado en políticas: BGP permite la implementación de políticas de enrutamiento flexibles y personalizadas para controlar el flujo de tráfico y tomar decisiones de enrutamiento basadas en criterios específicos.

Path Vector Protocol: BGP utiliza un enfoque de vector de ruta, lo que significa que no solo se tiene en cuenta la dirección IP de destino, sino también el camino o vector de ruta utilizado para llegar a la red de destino.

Escalabilidad: BGP está diseñado para redes grandes y complejas, y puede manejar una gran cantidad de rutas y actualizaciones de enrutamiento.

Soporte para múltiples direcciones de familia: BGP admite el intercambio de información de enrutamiento para diferentes familias de direcciones, como IPv4, IPv6, VPNv4, VPNv6, entre otras.

Toma de decisiones de enrutamiento basada en atributos: BGP utiliza atributos de enrutamiento para tomar decisiones sobre la mejor ruta para alcanzar una red de destino. Algunos de los atributos comunes son el valor del AS path, la longitud del prefijo, la preferencia local, entre otros.

Confiabilidad y resiliencia: BGP incluye mecanismos para garantizar la resiliencia de la red, como el uso de sesiones TCP (Puerto 179), detección y recuperación de fallas, y la capacidad de anunciar rutas alternativas en caso de fallas en la ruta principal.

Sumarización de rutas: BGP permite la sumarización de rutas para reducir la cantidad de información de enrutamiento que se intercambia.

Peering de ISP y Exchange de tráfico: BGP es ampliamente utilizado para establecer sesiones de peering entre proveedores de servicios de Internet (ISPs) y para el intercambio de tráfico en puntos de intercambio de Internet (IXPs).

Políticas de filtrado y control de tráfico: BGP ofrece flexibilidad en la aplicación de políticas de filtrado y control de tráfico para limitar el intercambio de rutas y controlar el flujo de tráfico dentro de la red.

 **BGP** Van Beijnum

 **JUNOS**
Enterprise Routing Marschke & Reynolds

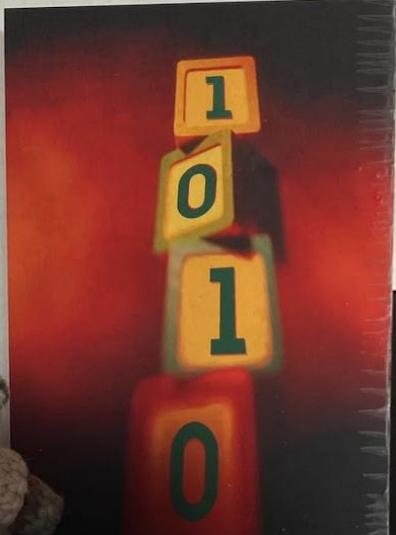
 **CISCO** CCIE
CCIE Professional Development
Routing TCP/IP
Volume I, Second Edition Doyle Carroll

 **CISCO** CCIE
Doyle
CCIE Professional Development
Routing TCP/IP
Volume II
Second Edition

 **CISCO**
Troubleshooting BGP

BGP4
Inter-Domain Routing in the Internet

JOHN W. STEWART III



THE ADDISON-WESLEY NETWORKING BASICS SERIES



BGP – Routing Algorithm*

*According to RFC4271 – Implementations are vendor-specific

1. Check if *next hop* is reachable
- 2. Choose route with the highest **Local Preference**
- 3. Prefer the route with the shortest **AS path**
4. Prefer the route with the lowest *origin attribute*
- 5. Prefer the route with the lowest **MED** value
6. Prefer routes received from *eBGP* over *iBGP*
7. Prefer the nearest *exit* from your network (in terms of your internal routing protocol)
- 8. **Implementation dependent:**
Prefer older (= more stable) routes
9. Prefer routes learned from the router with lower *router ID*
10. Prefer routes learned from the router with lower *IP address*

This is where you prefer peering over upstream

Next hop reachable?	Continue if "yes"
Local Preference	higher wins
AS path	shorter wins
Origin Type	IGP over EGP over inco
MED	lower wins
eBGP, iBGP	eBGP wins
Network exit	nearest wins
Age of route	older wins
Router ID	lower wins
Neighbor IP	lower wins

→ = most important rules

Le encantan las naranjas ...

We Love Oranges AS Oranges Mean Pure Refreshment



W	Weight (+Alto)
L	LOCAL_PREF (+ Alto)
O	Originate (local)
AS	AS_PATH (+ corto)
O	ORIGIN Code (IGP - EGP - Incomplete)
M	MED (+ bajo)
P	Paths (External - Internal)
R	Router_ID (+ bajo)

BGP - RFC

RFC 4271: "A Border Gateway Protocol 4 (BGP-4)"

RFC 4456: "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)"

RFC 4760: "Multiprotocol Extensions for BGP-4"

RFC 4272: "BGP Security Vulnerabilities Analysis"

RFC 1997: "BGP Communities Attribute"

RFC 8092: "BGP Large Communities Attribute"

RFC 6286: "Autonomous-System-Wide Unique BGP Identifier for BGP-4"

RFC 6608: "Subcodes for BGP Cease Notification Message"



Pocket RFC (by Okeno Palmer)
(Lector RFC en el móvil)

El tiempo pasa

Y la casa sin barrer



<https://chat.openai.com/>



A “Trabajar” !!!

CA

Puedes ponerte como si fueses un experto en networking



¡Por supuesto! Como experto en redes, puedo ayudarte con preguntas y conceptos relacionados con networking. No dudes en preguntar cualquier cosa específica que necesites saber o en solicitar información sobre temas como protocolos de enrutamiento, arquitecturas de red, seguridad de redes, tecnologías de red, y mucho más. Estoy aquí para ayudarte en lo que necesites.





puedes configurar una sesión BGP entre dos Peers BGP para un equipo arista



Por supuesto, aquí tienes un ejemplo de configuración de una sesión BGP entre dos peers BGP en un equipo Arista:

```
php Copy code

! Configuración global de BGP
router bgp <número-de-AS>
  router-id <dirección-IP-del-router>
  maximum-paths <número-de-caminos>
  neighbor <dirección-IP-del-peer> remote-as <número-de-AS-del-peer>
  neighbor <dirección-IP-del-peer> password <contraseña-del-peer>
  neighbor <dirección-IP-del-peer> maximum-routes <número-máximo-de-rutas>

! Configuración de anuncios de red
network <prefijo-de-red-a-anunciar>

! Configuración de políticas de enrutamiento
neighbor <dirección-IP-del-peer> route-map <nombre-de-política> in
neighbor <dirección-IP-del-peer> route-map <nombre-de-política> out
```

Explicación de los comandos:

- `<número-de-AS>`: Reemplaza esto con el número de sistema autónomo (ASN) de tu propio sistema.
- `<dirección-IP-del-router>`: Especifica la dirección IP del router Arista.
- `<número-de-caminos>`: Define el número máximo de caminos permitidos para el enrutamiento.
- `<dirección-IP-del-peer>`: Indica la dirección IP del peer BGP con el que deseas establecer la sesión.
- `<número-de-AS-del-peer>`: Configura el número de sistema autónomo del peer BGP.
- `<contraseña-del-peer>`: Establece una contraseña opcional para la autenticación del peer BGP.
- `<número-máximo-de-rutas>`: Establece el número máximo de rutas que se pueden recibir del peer.
- `<prefijo-de-red-a-anunciar>`: Inserta el prefijo de red que deseas anunciar al peer BGP.
- `<nombre-de-política>`: Indica el nombre de la política de enrutamiento utilizada para filtrar o modificar las rutas.



puedes configurar una sesión BGP entre dos peers BGP para un router Juniper



¡Claro! Aquí tienes un ejemplo de configuración de una sesión BGP entre dos peers BGP en un router Juniper. Asegúrate de adaptar los valores según tus necesidades de red.

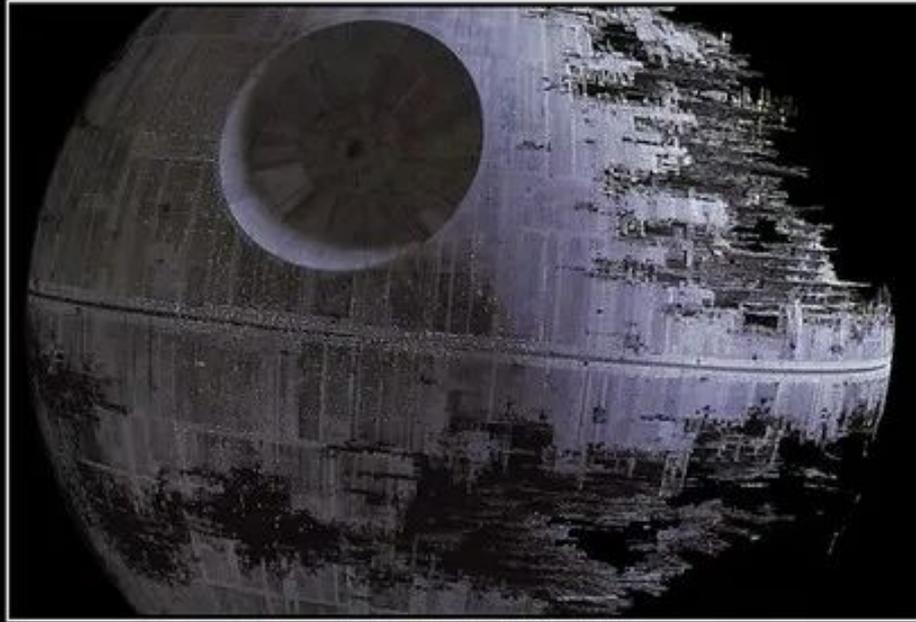
```
php Copy code

type external
neighbor <dirección-IP-del-peer> peer-as <número-de-AS-del-peer>
neighbor <dirección-IP-del-peer> authentication-key <contraseña-del-peer>
neighbor <dirección-IP-del-peer> family inet unicast
neighbor <dirección-IP-del-peer> export <nombre-de-política-de-exportación>
neighbor <dirección-IP-del-peer> import <nombre-de-política-de-importación>
neighbor <dirección-IP-del-peer> local-as <número-de-AS>
neighbor <dirección-IP-del-peer> hold-time <valor-de-tiempo-de-espera>
```

Explicación de los comandos:

- `<nombre-del-grupo>`: Especifica un nombre para el grupo BGP.
- `<dirección-IP-del-peer>`: Indica la dirección IP del peer BGP con el que deseas establecer la sesión.
- `<número-de-AS-del-peer>`: Configura el número de sistema autónomo del peer BGP.
- `<contraseña-del-peer>`: Establece una contraseña opcional para la autenticación entre los peers.
- `<nombre-de-política-de-exportación>`: Define el nombre de la política de exportación que controla las rutas a anunciar al peer.
- `<nombre-de-política-de-importación>`: Especifica el nombre de la política de importación que controla las rutas aceptadas del peer.
- `<número-de-AS>`: Configura el número de sistema autónomo (ASN) de tu propio sistema.
- `<valor-de-tiempo-de-espera>`: Establece el tiempo de espera (hold time) para la sesión BGP.

Recuerda que también necesitarás configurar las interfaces de los routers con las direcciones IP correspondientes y asegurarte de que haya conectividad entre ellos antes de configurar BGP. Además, ten en cuenta que esta es solo una configuración básica y puede haber otras opciones y ajustes adicionales dependiendo de tus requisitos de red.



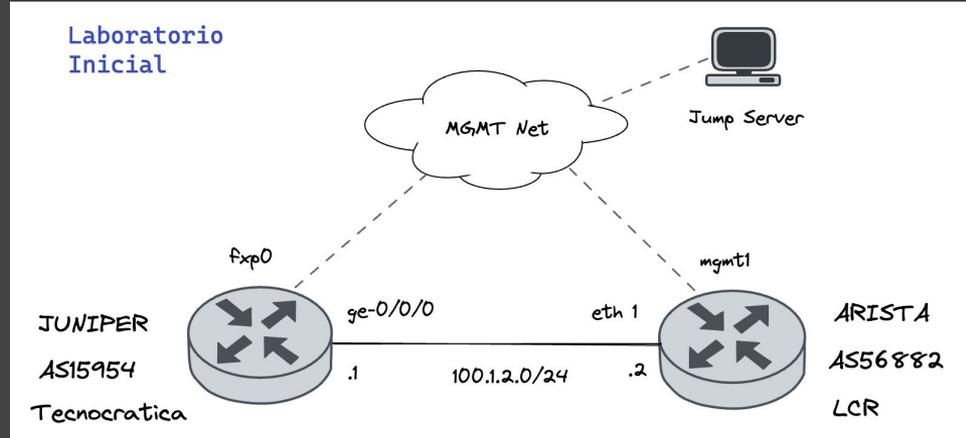
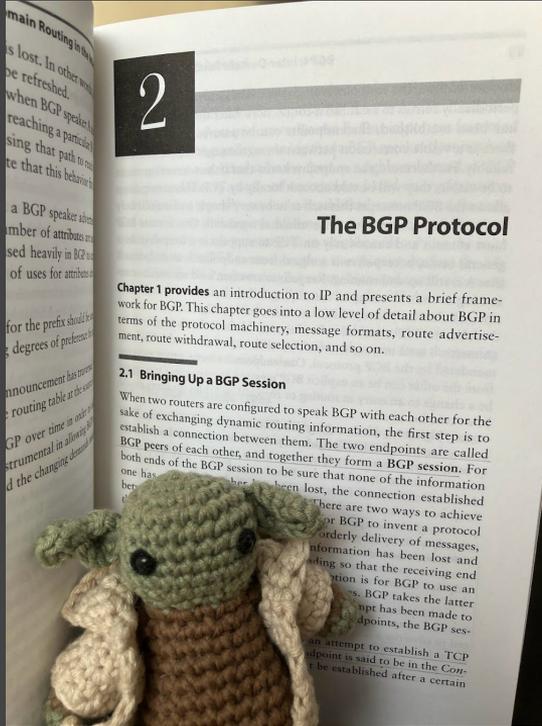
TESTING

DON'T BE TOO PROUD OF THIS TECHNOLOGICAL TERROR YOU'VE
CONSTRUCTED. THE ABILITY TO DESTROY A PLANET IS INSIGNIFICANT NEXT
TO THE POWER OF TESTING.

DYZDESIGN.COM

TESTING: No estés demasiado orgulloso de este terror tecnológico que has construido. La capacidad de destruir un planeta es insignificante al lado del poder de prueba

Laboratorio



* Pido disculpas por ser IPv4 only 🙏🙏🙏🙏

router bgp 56882

router-id 100.1.2.2

neighbor 100.1.2.1 remote-as 15954

neighbor 100.1.2.1 password ME_GUSTA_EL_CAFE

neighbor 100.1.2.2 update-source interface Ethernet5

network 185.183.8.0/22

```
lcr-as56882#sh ip bgp summary
BGP summary information for VRF default
Router identifier 100.1.2.2, local AS number 56882
Neighbor Status Codes: m - Under maintenance
Neighbor      V  AS      MsgRcvd   MsgSent   InQ  OutQ  Up/Down  State       PfxRcd  PfxAcc
100.1.2.1     4  15954     4         4         0    0  00:00:15  Estab       0       0
```

(*) la configuración de las interfaces en ambos casos las hace perfectamente

set protocols bgp group PEER-BGP type external

set protocols bgp group PEER-BGP neighbor 100.1.2.2 peer-as 56882

set protocols bgp group PEER-BGP neighbor 100.1.2.2 authentication-key ME_GUSTA_EL_CAFE

set protocols bgp group PEER-BGP neighbor 100.1.2.2 family inet unicast

set protocols bgp group PEER-BGP neighbor 100.1.2.2 local-as 15954

```
root@tecnocratica-as15954> show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths  Suppressed    History  Damp State    Pending
inet.0
Peer          AS      InPkt    OutPkt    OutQ   Flaps Last Up/Dwn  State|#Active/Received/Accepted/Damped...
100.1.2.2    56882     6         6         0       0    1:46  0/0/0/0      0/0/0/0
```

No vemos los prefijos anunciados desde el Arista en el Juniper

```
root@tecnocratica-as15954> show route receive-protocol bgp 100.1.2.2
inet.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
```

Troubleshooting →



a pesar que he puesto el comando `network <prefijo-de-red-a-anunciar>` no veo las redes anunciadas en el otro peer, a que puede ser debido



Si has configurado correctamente el comando `network <prefijo-de-red-a-anunciar>` en BGP, pero no ves las redes anunciadas en el otro peer, aquí hay algunas posibles razones:

- Verificación de las políticas de enrutamiento:** Asegúrate de que las políticas de enrutamiento en el peer receptor permitan el anuncio de las redes que estás intentando anunciar. Revisa las configuraciones de las listas de acceso, las listas de filtros o las políticas de importación y exportación utilizadas en el peer receptor.
- Verificación de la conectividad:** Asegúrate de que haya conectividad entre los dos peers BGP. Verifica que las direcciones IP y las configuraciones de las interfaces de los routers estén correctamente configuradas y que haya conectividad de red adecuada entre ellos.
- Verificación del estado de la sesión BGP:** Comprueba si la sesión BGP entre los dos peers está establecida correctamente. Puedes utilizar comandos como `show bgp summary` o `show bgp neighbor` en los routers para verificar el estado de la sesión BGP y asegurarte de que esté en el estado "Established".
- Verificación de los filtros y políticas de importación:** Si estás utilizando filtros o políticas de importación en el peer receptor, verifica si estas políticas están filtrando las redes que intentas anunciar. Asegúrate de que las políticas estén configuradas adecuadamente para permitir el anuncio de las redes deseadas.
- Verificación de la tabla de enrutamiento:** Comprueba la tabla de enrutamiento en el peer receptor para asegurarte de que las rutas anunciadas por el peer origen estén presentes. Utiliza comandos como `show ip route` o `show bgp routes` en el peer receptor para verificar si las rutas están siendo recibidas correctamente.

Estas son solo algunas posibles razones por las que las redes anunciadas no se están reflejando en el otro peer. Recuerda que la solución específica dependerá de tu configuración y de la plataforma de enrutamiento que estás utilizando.

Si sigues teniendo problemas, proporciona más detalles sobre tu configuración específica y los comandos de verificación que has utilizado, para que pueda ayudarte de manera más precisa.



aquí tienes mi configuración de routing en el Arista, para que puedas analizarlo por si falta algún comando:

Config Arista to chat GPT

Route to null_0

```
ip route 185.183.8.0/22 null0
```

Juniper:

```
show route receive-protocol bgp 100.1.2.2
```

```
inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
  Prefix            Nexthop          MED      Lclpref   AS path
* 185.183.8.0/22    100.1.2.2        0         0         56882 ?
```



Gracias por proporcionar tu configuración. Parece que has configurado la sesión BGP correctamente en tu router Arista. Sin embargo, hay un detalle en la configuración que podría estar causando el problema.

En la línea "network 185.183.8.0/22", estás intentando anunciar la red 185.183.8.0/22 a través de BGP. Sin embargo, en tu configuración de interfaz, solo tienes asignada la dirección IP 100.1.2.2/24 a la interfaz Ethernet5. Para que la red 185.183.8.0/22 se anuncie correctamente a través de BGP, debes asegurarte de que la interfaz Ethernet5 esté conectada a la red 185.183.8.0/22.

Si la interfaz Ethernet5 está conectada a la red 185.183.8.0/22, entonces el problema puede estar en la configuración del peer BGP remoto (dirección IP 100.1.2.1). Asegúrate de que el peer BGP remoto esté configurado correctamente y esté en el estado "Established". También verifica las políticas de enrutamiento en el peer remoto para asegurarte de que esté permitiendo las rutas anunciadas desde tu router Arista.

Si todo está configurado correctamente y sigues teniendo problemas, es posible que debas verificar los registros o mensajes de error del router Arista para obtener más detalles sobre el problema específico.

Su responsable ha quedado muy satisfecho del progreso que ha hecho en un único fin de semana.

Consecuencia:

Configurar más peers BGP (en este caso contra RIPE)

Volvemos al laboratorio, GruBuP necesita más de confianza con el CLI

Os recuerdo que ... le encanta beber café



Consecuencia: AUTOMATIZACIÓN

CA

como puedo automatizar esta configuración BGP para futuros peers



Para automatizar la configuración BGP para futuros peers, puedes utilizar herramientas de automatización de red como Ansible, Python o herramientas específicas de gestión de configuración de dispositivos de red. Estas herramientas te permiten definir plantillas de configuración y aplicarlas de manera programática en múltiples dispositivos.

A continuación, te mostraré un ejemplo de cómo podrías utilizar Ansible para automatizar la configuración BGP en múltiples peers:

1. Instala Ansible en tu máquina de control.
2. Crea un inventario de hosts que contenga los detalles de los dispositivos (dirección IP, credenciales, etc.) donde deseas aplicar la configuración BGP.
3. Define una plantilla de configuración BGP utilizando el formato de configuración específico del dispositivo (por ejemplo, formato YAML para Juniper o formato Jinja2 para Cisco).
4. Crea un playbook de Ansible que utilice la plantilla de configuración BGP y aplícala en los hosts del inventario.
5. Ejecuta el playbook de Ansible para aplicar automáticamente la configuración BGP en todos los peers especificados en el inventario.

Aquí tienes un ejemplo básico de cómo podría verse un playbook de Ansible para configurar BGP en un dispositivo Juniper:

yaml

Copy code

```
- name: Configurar BGP en dispositivos Juniper
  hosts: bgp_peers
  gather_facts: no
  tasks:
    - name: Cargar la plantilla de configuración BGP
      template:
        src: bgp_template.j2
        dest: /ruta/de/destino/bgp_config.conf
      notify: Reiniciar BGP

    - name: Copiar el archivo de configuración al dispositivo
      junos_command:
        commands: ["request system configuration rescue save"]
      provider: "{{ credentials }}"
```

En este ejemplo, `bgp_peers` es el grupo de hosts definido en el inventario que contiene los detalles de los peers BGP. `bgp_template.j2` es la plantilla de configuración BGP en formato Jinja2 que se utilizará para generar la configuración específica de cada dispositivo.

Recuerda que este es solo un ejemplo básico y deberás adaptarlo según tus necesidades específicas. También es importante considerar la seguridad y las mejores prácticas al automatizar la configuración de tus dispositivos de red.

Espero que esto te ayude a automatizar la configuración BGP en futuros peers. Si tienes más preguntas, no dudes en preguntar.



me puedes mostrar la configuración del archivo `bgp_template.j2`



Por supuesto. Aquí tienes un ejemplo de cómo podría ser la configuración en la plantilla `'bgp_template.j2'` para un dispositivo Juniper:

```
sql Copy code

protocols {
  bgp {
    group {{ bgp_group_name }} {
      type external;
      neighbor {{ peer_ip_address }} {
        peer-as {{ peer_as_number }};
        authentication-key "{{ peer_password }}";
        family inet unicast;
        export {{ export_policy }};
        import {{ import_policy }};
        local-as {{ local_as_number }};
        hold-time {{ hold_time }};
      }
    }
  }
}
```

En este ejemplo, utilizamos variables dentro de la plantilla `'{{ }}'` para los valores que pueden variar según cada peer BGP. A continuación, se explica el significado de cada línea de la plantilla:

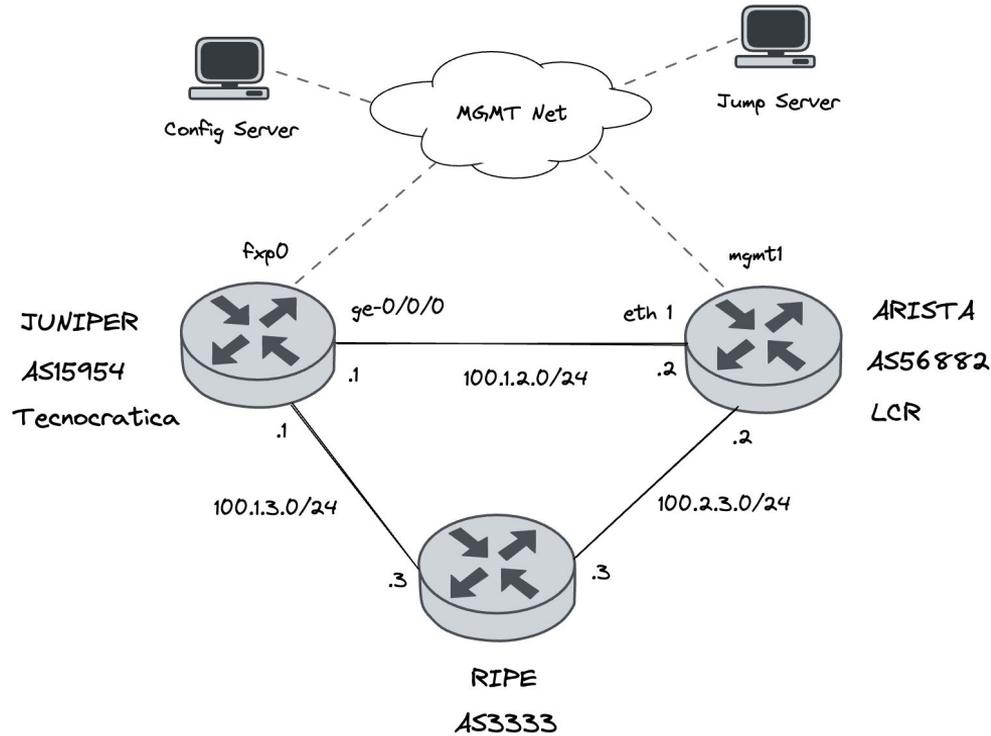
- `'bgp_group_name'`: Variable que representa el nombre del grupo BGP.
- `'peer_ip_address'`: Variable que representa la dirección IP del peer BGP.
- `'peer_as_number'`: Variable que representa el número de sistema autónomo (ASN) del peer BGP.
- `'peer_password'`: Variable que representa la contraseña de autenticación del peer BGP.
- `'export_policy'`: Variable que representa el nombre de la política de exportación utilizada para controlar las rutas a anunciar al peer.
- `'import_policy'`: Variable que representa el nombre de la política de importación utilizada para controlar las rutas aceptadas del peer.
- `'local_as_number'`: Variable que representa el número de sistema autónomo (ASN) local.
- `'hold_time'`: Variable que representa el tiempo de espera (hold time) para la sesión BGP.

Recuerda que estas variables deberán ser definidas y pasadas desde el playbook de Ansible al renderizar la plantilla, para que los valores correspondientes se inserten en la configuración generada para cada peer BGP.

Ten en cuenta que este es solo un ejemplo básico y que la plantilla de configuración puede ser más compleja dependiendo de tus necesidades y la sintaxis específica del dispositivo Juniper.

Espero que esto te sea útil. Si tienes más preguntas, no dudes en preguntar.

Laboratorio Automatización



Arista con ChatGPT

GruBuP hace caso a chatGPT y le dice que lo haga con Ansible.

Probamos un simple playbook en el laboratorio:

```
• carlos@confsrv01:~/ansible$ ansible-playbook arista-inventory.yml -i inventory/esnog29

PLAY [SHOW SERIAL NUMBER] *****

TASK [GATHERING INFO] *****
ok: [lcr-as56882]

TASK [Hostname - Model (P/N) - Serial Number] *****
ok: [lcr-as56882] => {
  "msg": "lcr-as56882 - vEOS-lab - BBA84767804F9CDD12A1F68BB9C41C0F"
}

PLAY RECAP *****
lcr-as56882      : ok=2   changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

```
lcr-as56882#sh ver
Arista vEOS-lab
Hardware version:
Serial number: BBA84767804F9CDD12A1F68BB9C41C0F
Hardware MAC address: 5000.0088.fe27
System MAC address: 5000.0088.fe27

Software image version: 4.26.5M
Architecture: i686
Internal build version: 4.26.5M-26324336.4265M
Internal build ID: 9f99e3a9-be90-42ee-af19-b67dfec942dc
Image format version: 1.0

Uptime: 2 hours and 49 minutes
Total memory: 2006804 kB
Free memory: 1307228 kB
```

Parece que funciona, vamos a por BGP con el café en la mesa !!!



¿me puedes generar un fichero yml en ANSIBLE para configurar una sesion BGP en un equipo ARISTA



¡Claro! Aquí tienes un ejemplo de un archivo YAML (fichero yml) para configurar una sesión BGP en un equipo Arista utilizando Ansible:

```

yaml
---
- name: Configurar sesión BGP en equipo Arista
  hosts: arista_devices
  gather_facts: no
  connection: network_cli

  vars:
    bgp_local_as: 65000
    bgp_remote_as: 65001
    bgp_neighbor: 192.168.1.1

  tasks:
    - name: Configurar BGP
      arista.eos.bgp:
        local_as: "{{ bgp_local_as }}"
        peer_group: "{{ bgp_remote_as }}"
        peer: "{{ bgp_neighbor }}"
        password: "mypassword"
        state: present
  
```

En este ejemplo, se configura una sesión BGP en un equipo Arista utilizando el módulo `'arista.eos.bgp'` de Ansible. Aquí se asume que tienes el módulo instalado en tu entorno Ansible.

Asegúrate de personalizar las variables según tu entorno:

- `'bgp_local_as'`: El número de Sistema Autónomo local (AS) utilizado para BGP.
- `'bgp_remote_as'`: El número de Sistema Autónomo remoto (AS) del vecino BGP.
- `'bgp_neighbor'`: La dirección IP del vecino BGP.

Puedes ejecutar este playbook utilizando el siguiente comando:

```

css
ansible-playbook -i inventory_file.yml bgp_configuration.yml
  
```

```

TASK [Configurar BGP] *****
task path: /home/carlos/ansible/arista-bgp_v0.yml:29
redirecting (type: terminal) ansible.builtin.eos to arista.eos.eos
redirecting (type: cliconf) ansible.builtin.eos to arista.eos.eos
[DEPRECATION WARNING]: arista.eos.bgp has been deprecated. See the plugin documentation for more details. This feature will be removed from arista.eos in a release after 2023-01-01. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
redirecting (type: modules) arista.eos.bgp to arista.eos.eos_bgp
[DEPRECATION WARNING]: arista.eos.eos_bgp has been deprecated. See the plugin documentation for more details. This feature will be removed from arista.eos in a release after 2023-01-01. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
ansible.cfg:
redirecting (type: modules) arista.eos.bgp to arista.eos.eos_bgp
redirecting (type: modules) arista.eos.bgp to arista.eos.eos_bgp
fatal: [1cr-as5682]: FAILED! => ('change!': false, 'bug': 'Unsupported parameters for (arista.eos.bgp) module: interface, local_as, password, peer, peer_group, state. Supported parameters include: config, operation.')
  
```

```

carlos@confsv01:~/ansible$ ansible --version
ansible [core 2.14.5]
  config file = /home/carlos/ansible/ansible.cfg
  configured module search path = ['/home/carlos/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/carlos/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.10.6 (main, Mar 10 2023, 10:55:28) [GCC 11.3.0] (/usr/bin/python3)
  jinja version = 3.0.3
  libyaml = True
  
```

Lamento la confusión anterior. Parece que cometí un error en el ejemplo proporcionado.

Sigue sin funcionar ... Nos toca buscar por Internet

Problemas

arista.eos.bgp module

Note

This redirect is part of the `arista.eos` collection (version 6.0.0).

- This redirect has been **deprecated**. Please update your tasks to use the new name `arista.eos.eos_bgp` instead. It will be removed in a major release after 2023-01-01 of `arista.eos`.

Salimos de chatGPT y miramos en la documentación de Ansible nos confirma que está “deprecated”

https://docs.ansible.com/ansible/latest/collections/arista/eos/eos_bgp_global_module.html

Le pedimos a chatGPT que lo haga de nuevo con un módulo actualizado pero todos los playbooks que nos ofrece fallan una y otra vez, lo va corrigiendo sobre la marcha.

Muchas veces vuelve al módulo que está obsoleto. Una y otra vez

```
carlos@confsrv01:~/ansible$ yamllint arista-bgp_v0.yml
arista-bgp_v0.yml
20:3 error wrong indentation: expected 4 but found 2 (indentation)
30:5 error syntax error: expected <block end>, but found '-' (syntax)
39:81 error line too long (92 > 80 characters) (line-length)
42:1 error too many blank lines (2 > 0) (empty-lines)
```

Al copiar el fichero .yml tiene errores que gracias a yamllint podemos corregir.

Después de 4 horas ... No llegamos a establecer ninguna sesión BGP con Ansible mediante las configuraciones que nos ofrece chatGPT, lo dejamos y nos vamos hacer un café que hay vida detrás de la pantalla.

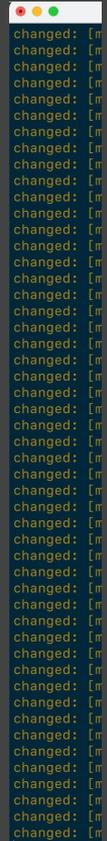
Quizás nosotros cometimos algún error, supongo que alguien ha tenido más suerte y le funciona.

Disclaimer

Que no le funcionen los playbooks de Ansible a chatGPT no significa que no funcione en la vida real.

En mi día a día toco Arista: Y no los cambio por nada del mundo

Los playbooks en Ansible, scripts en Python funcionan perfectamente !!!



Juniper con ChatGPT

GruBuP hablando con un compañero durante el café le recomienda Python

CA puedes ponerte en modo experto en network, juniper y python

CA necesito un programa en python utilizando la netconf para configurar una sesion BGP con los siguiente datos:

🌐 Aquí tienes un ejemplo de un programa en Python utilizando la biblioteca ncclient para configurar una sesión BGP utilizando NETCONF en un enrutador Juniper, con los datos proporcionados:

```
root@ROU-01> show version
Hostname: ROU-01
Model: vmx
Junos: 14.1R1.10
JUNOS Base OS Software Suite [14.1R1.10]
JUNOS Base OS boot [14.1R1.10]
JUNOS Crypto Software Suite [14.1R1.10]
JUNOS Online Documentation [14.1R1.10]
JUNOS Kernel Software Suite [14.1R1.10]
JUNOS Packet Forwarding Engine Support (M320) [14.1R1.10]
JUNOS Packet Forwarding Engine Support (M/T/EX Common) [14.1R1.10]
```

Una cosa que aprendió:

- Especificar la versión

Después de leer sobre las diferentes librerías, cómo instalarlas y para qué sirve cada una de ellas:

adapta el código para Junos:14.1R1.10

🌐 La versión de Junos OS 14.1R1.10 es bastante antigua y puede tener diferencias en la sintaxis y los elementos de configuración en comparación con las versiones más recientes. A continuación, se muestra una adaptación del código para utilizar la sintaxis compatible con Junos OS 14.1R1.10:

```
python
from ncclient import manager
from jnpr.junos import Device
from jnpr.junos.utils.config import Config

# Datos de conexión
hostname = "ROU-01"
ip_address = "192.168.1.1" # Dirección IP del enrutador Juniper
username = "tu_usuario"
password = "tu_contraseña"

# Datos de configuración BGP
bgp_local_as = "15954"
bgp_remote_as = "3333"
bgp_peer_ip = "100.1.3.3"
bgp_password = "ME_GUSTA_EL_CAFE"

# Configuración BGP
config_template = """
set system host-name (hostname)
set protocols bgp group mygroup type external
set protocols bgp group mygroup local-as (bgp_local_as)
set protocols bgp group mygroup neighbor (bgp_peer_ip) peer-as (bgp_remote_as)
set protocols bgp group mygroup neighbor (bgp_peer_ip) authentication-key (
"""

# Conexión al enrutador Juniper
with Device(host_ip_address, user=username, password=password) as dev:
    # Crear objeto de configuración
    cu = Config(dev)

    # Abrir una transacción de configuración
    cu.lock()
```

Juniper con ChatGPT

¿Qué ha pasado ?

```
● carlos@confsrv01:~/scripts/python$ python3 juniper_bgp.py
○ carlos@confsrv01:~/scripts/python$
```

¿ Se ha configurado, está establecida ?

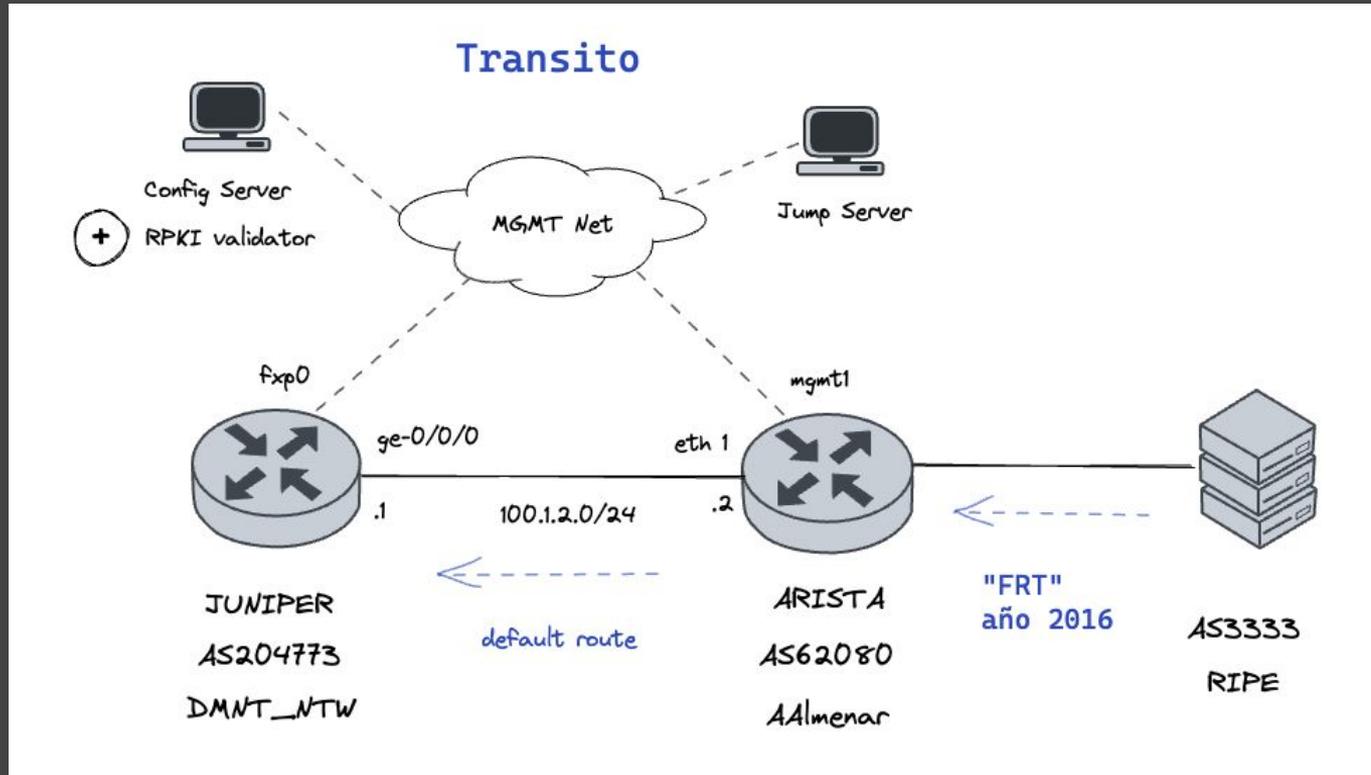
```
root@tecnocratica-as15954> show configuration protocols bgp
group PEER-BGP {
  type external;
  neighbor 100.1.2.2 {
    family inet {
      unicast;
    }
    authentication-key "$S9$77-w2Z0jfznu0NVY2ZG90tu01Lx-Y2omfMLN-sYg4aGDHF5z"; ## SECRET-DATA
    peer-as 56882;
    local-as 15954;
  }
}
group mygroup {
  type external;
  local-as 15954;
  neighbor 100.1.3.3 {
    authentication-key "$S9$D5i.f3n9BESWLjkPf3n1evW8xZUiPfQ0BoZjimP5Tzn/AB1E"; ## SECRET-DATA
    peer-as 3333;
  }
}
```

```
root@tecnocratica-as15954> show bgp summary
Groups: 2 Peers: 2 Down peers: 0
Table Tot Paths Act Paths Suppressed History Damp State Pending
inet.0 1 1 0 0 0 0 0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn State|#Active/Received/Accepted/Damped...
100.1.2.2 56882 188 207 0 0 1:32:36 1/1/1/0 0/0/0/0
100.1.3.3 3333 171 190 0 0 1:24:37 0/0/0/0 0/0/0/0
```

Tránsito y Peering

- Configuración sesión BGP (re-aprovechar código del Juniper)
- Toca Aprender cosas nuevas:
 - Filtrado Bogons y Martians
 - Generación de Prefix-list de forma automática
 - Filtrado AS-PATH
 - Firmar RPKI
 - Validar RPKI
 - MANRS

Configuración tránsito



Transito y Peering

La sesión del juniper → Aprovechamos código anterior

```
BGP summary information for VRF default
Router identifier 100.1.2.2, local AS number 62080
Neighbor Status Codes: m - Under maintenance
Neighbor      V AS      MsgRcvd  MsgSent  InQ  OutQ  Up/Down State  PfxRcd  PfxAcc
100.0.1.1     4 3333    1307190    76    0    0 00:01:54 Estab  1866   1866
100.1.2.1     4 204773     69    90859    0    0 00:03:00 Estab    0     0
```

Preguntamos para enviar una default route a nuestro peer (Juniper)

CA yo soy un operador de tránsito y quiero anunciar una ruta por defecto a un único peer BGP que es cliente mío, en vez de anunciar una full routing table, que configuración debería tener en un equipo Arista

Como operador de tránsito, si deseas anunciar una ruta por defecto a un único cliente BGP en lugar de anunciar una tabla de enrutamiento completa, puedes seguir estos pasos en un equipo Arista:

1. Accede al modo de configuración de Arista:

```
bash
enable
configure terminal
```

2. Configura una ruta por defecto hacia el próximo salto:

```
php
ip route 0.0.0.0/0 <next_hop_ip_address>
```

Donde `<next_hop_ip_address>` es la dirección IP del próximo salto que utilizará la ruta por defecto. Esta dirección IP debe ser accesible y estar conectada directamente a tu equipo Arista.

3. Configura el anuncio de la ruta por defecto hacia el peer BGP específico:

```
php
router bgp <your_as_number>
  neighbor <peer_ip_address> remote-as <peer_as_number>
  neighbor <peer_ip_address> route-map announce-default out
```

Donde `<your_as_number>` es tu número de sistema autónomo (AS), `<peer_ip_address>` es la dirección IP del cliente BGP al que deseas enviar la ruta por defecto y `<peer_as_number>` es el número de sistema autónomo del cliente BGP.

4. Configura un route-map para filtrar las rutas que se anunciarán al peer BGP:

```
arduino
route-map announce-default permit 10
  match ip address prefix-list default-route
```

5. Configura una lista de prefijos (prefix-list) para incluir solo la ruta por defecto:

```
swift
ip prefix-list default-route seq 5 permit 0.0.0.0/0
```

Esto asegurará que solo se anuncie la ruta por defecto hacia el cliente BGP específico y no se propague la tabla de enrutamiento completa.

que seguridad en si mismo !!!

Pues vamos a configurar !!!

Configuramos y revisamos el Juniper:

```
root@dmnt_ntw-as204773> show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table
inet.0
Peer      Tot Paths  Act Paths  Suppressed  History Damp State  Pending
100.1.2.2  12037     12037      0            0        0      0
AS        InPkt     OutPkt     OutQ      Flaps  Last Up/Dwn  State|#Active/Received/Accepted/Damped...
62080    22624     42         0          1      18:02  12037/12037/12037/0 0/0/0/0
```

```
root@dmnt_ntw-as204773> show bgp group PEER-BGP detail
Group Type: External      Local AS: 204773
Name: PEER-BGP           Index: 0          Flags: <Export Eval>
Options: <LocalAS>
Holdtime: 0 Local AS: 204773 Local System AS: 0
Total peers: 1           Established: 1
100.1.2.2+37603
Route Queue Timer: unset Route Queue: empty
Table inet.0
Active prefixes:         5183
Received prefixes:      5183
Accepted prefixes:      5183
Suppressed due to damping: 0
Advertised prefixes:    0
```

```
root@dmnt_ntw-as204773> show route protocol bgp
inet.0: 7067 destinations, 7067 routes (7067 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0 * [BGP/170] 00:09:28, localpref 100
AS path: 62080 ?, validation-state: unverified
> to 100.1.2.2 via ge-0/0/3.0
1.0.0.0/24 * [BGP/170] 00:01:49, localpref 100
AS path: 62080 3333 8758 8220 15169 I, validation-state: unverified
> to 100.1.2.2 via ge-0/0/3.0
1.0.4.0/24 * [BGP/170] 00:01:49, localpref 100
AS path: 62080 3333 8758 8758 6939 4826 38803 56203 I, validation-state: unverified
> to 100.1.2.2 via ge-0/0/3.0
1.0.5.0/24 * [BGP/170] 00:01:48, localpref 100
AS path: 62080 3333 8758 8758 6939 4826 38803 56203 I, validation-state: unverified
> to 100.1.2.2 via ge-0/0/3.0
1.0.6.0/24 * [BGP/170] 00:01:48, localpref 100
AS path: 62080 3333 8758 174 4826 38803 56203 56203 I, validation-state: unverified
> to 100.1.2.2 via ge-0/0/3.0
1.0.7.0/24 * [BGP/170] 00:01:48, localpref 100
AS path: 62080 3333 8758 8220 4637 1221 38803 I, validation-state: unverified
> to 100.1.2.2 via ge-0/0/3.0
```

Filtrado Bogons / Martians



que diferencia hay entre un prefijo BOGON y un MARTIAN



Tanto los prefijos BOGON como los prefijos MARTIAN son términos utilizados en el contexto de direcciones IP para referirse a rangos específicos que se consideran inválidos o no ruteables en Internet. Sin embargo, hay una diferencia fundamental entre ellos:

1. **Prefijos BOGON:** Los prefijos BOGON son rangos de direcciones IP que no están asignados o son reservados para uso especial. Estos prefijos no deben aparecer en las tablas de enrutamiento de Internet ni ser anunciados en sesiones BGP. Los prefijos BOGON incluyen direcciones IP como las reservadas para uso privado (por ejemplo, 10.0.0.0/8, 192.168.0.0/16) y otras direcciones especiales como 0.0.0.0/8 o 127.0.0.0/8. Los prefijos BOGON se utilizan para filtrar y evitar que se propaguen en la red.
2. **Prefijos MARTIAN:** Los prefijos MARTIAN se refieren a direcciones IP que nunca deben aparecer en el tráfico de Internet público. Estas direcciones son consideradas "marcianas" debido a que son inválidas y no se originan en fuentes legítimas. Los prefijos MARTIAN incluyen direcciones IP reservadas como 0.0.0.0/32, 0.0.0.0/8, 127.0.0.0/8 y otras direcciones utilizadas para pruebas, experimentación o propósitos especiales. Los prefijos MARTIAN deben ser filtrados y descartados por los dispositivos de red para evitar problemas de enrutamiento y seguridad.

En resumen, la diferencia principal entre los prefijos BOGON y los prefijos MARTIAN radica en su propósito y en los rangos de direcciones IP que abarcan. Los prefijos BOGON se refieren a rangos no asignados o reservados, mientras que los prefijos MARTIAN son direcciones inválidas que nunca deben aparecer en el tráfico de Internet público. Ambos tipos de prefijos son importantes para mantener una red segura y eficiente al filtrar y descartar direcciones IP inapropiadas o no ruteables.



me puedes indicar una lista de BOGONS y MARTIAN de IPv4 junto con su RFC y el título de este RFC asociado, indicando si es MARTIAN O BOGON



Aquí tienes una lista de los BOGONS y MARTIAN de IPv4 junto con las referencias RFC, y se indica si son BOGONS o MARTIAN:

1. 0.0.0.0/8 (RFC 1700) - BOGON - "Assigned Numbers"
2. 10.0.0.0/8 (RFC 1918) - BOGON - "Address Allocation for Private Internets"
3. 100.64.0.0/10 (RFC 6598) - BOGON - "IANA-Reserved IPv4 Prefix for Shared Address Space"
4. 127.0.0.0/8 (RFC 5735) - MARTIAN - "Special Use IPv4 Addresses"
5. 169.254.0.0/16 (RFC 3927) - MARTIAN - "Dynamic Configuration of IPv4 Link-Local Addresses"
6. 172.16.0.0/12 (RFC 1918) - BOGON - "Address Allocation for Private Internets"
7. 192.0.0.0/24 (RFC 6890) - BOGON - "Special-Purpose IP Address Registries"
8. 192.0.2.0/24 (RFC 5737) - BOGON - "IPv4 Address Blocks Reserved for Documentation"
9. 192.88.99.0/24 (RFC 3068) - BOGON - "An Anycast Prefix for 6to4 Relay Routers"
10. 192.168.0.0/16 (RFC 1918) - BOGON - "Address Allocation for Private Internets"
11. 198.18.0.0/15 (RFC 2544) - BOGON - "Benchmarking Methodology for Network Interconnect Devices"
12. 198.51.100.0/24 (RFC 5737) - BOGON - "IPv4 Address Blocks Reserved for Documentation"
13. 203.0.113.0/24 (RFC 5737) - BOGON - "IPv4 Address Blocks Reserved for Documentation"
14. 224.0.0.0/4 (RFC 3171) - MARTIAN - "IANA Guidelines for IPv4 Multicast Address Assignments"
15. 240.0.0.0/4 (RFC 1112) - MARTIAN - "Host Extensions for IP Multicasting"

Esta lista incluye tanto los rangos de direcciones BOGONS como los rangos MARTIAN de IPv4. Los BOGONS son rangos que no están asignados o son reservados, mientras que los MARTIAN son direcciones inválidas y especiales que no deben aparecer en el tráfico público de Internet. Los RFC asociados proporcionan las especificaciones y recomendaciones para el uso de estas direcciones IP. Es importante filtrar y descartar adecuadamente los BOGONS y MARTIAN en las configuraciones de red para mantener la seguridad y el correcto funcionamiento del enrutamiento.



me puedes generar una lista de acceso para un equipo ARISTA filtrando bogons y martians



Por supuesto, aquí tienes un ejemplo de cómo configurar una lista de acceso en un equipo Arista para filtrar los prefijos BOGON y MARTIAN:

```
arduino Copy code  
  
access-list bogons-martians  
10 deny ip 0.0.0.0/8 any  
20 deny ip 10.0.0.0/8 any  
30 deny ip 100.64.0.0/10 any  
40 deny ip 127.0.0.0/8 any  
50 deny ip 169.254.0.0/16 any  
60 deny ip 172.16.0.0/12 any  
70 deny ip 192.0.0.0/24 any  
80 deny ip 192.0.2.0/24 any  
90 deny ip 192.88.99.0/24 any  
100 deny ip 192.168.0.0/16 any  
110 deny ip 198.18.0.0/15 any  
120 deny ip 198.51.100.0/24 any  
130 deny ip 203.0.113.0/24 any  
140 deny ip 224.0.0.0/4 any  
150 deny ip 240.0.0.0/4 any  
160 permit ip any any
```

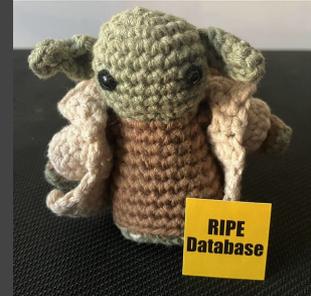
En este ejemplo, la lista de acceso "bogons-martians" deniega los prefijos BOGON y MARTIAN y permite cualquier otro tráfico. Asegúrate de aplicar esta lista de acceso en las interfaces relevantes de tu equipo Arista utilizando el comando "ip access-group bogons-martians in" en la interfaz correspondiente.

Recuerda que esta configuración puede variar dependiendo del modelo y la versión de software de tu equipo Arista, por lo que es importante adaptarla según tus necesidades y las características de tu dispositivo.

Source of truth: RIPE NCC Database

Es importante tener bien definido los objetos en RIPE*:

- Inetnum
- Domain objects
- Route objects



RPKI: Firmad vuestros ROA (Route Origin Authorisation)

Prohibido perderse esta charla a las 16:30 → Álvaro Vives (RIPE NCC): Routing Security

* Ripe Academy: RIPE Database

RPKI - Firmar ROA (Route Origin Authorisation)

LIR Portal

You are editing [redacted]

Create a Certificate Authority for [redacted]

RIPE NCC Certification Service Terms and Conditions

Introduction

This document will stipulate the Terms and Conditions for the RIPE NCC Certification Service. The RIPE NCC Certification Service is based on Internet Engineering Task Force (IETF) standards, in particular RFC3647, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework", RFC3779, "X.509 Extensions for IP Addresses and AS Identifiers", and the "Certificate Policy (CP) for the Resource PKI (RPKI)".

Article 1 – Definitions

Type of Certificate Authority

You can choose between asking the RIPE NCC to host your RPKI Certificate Authority (Hosted RPKI) or running your own Certificate Authority (Delegated RPKI).

Select "Hosted" if you would like the RIPE NCC to host your Certificate Authority, keys, ROAs, manifests etc. and publish the information in our repository. You will only need to maintain your ROAs in our dashboard. This is the recommended option if you are not an RPKI expert.

Select "Delegated" to run your own Certificate Authority and to host your own keys, ROAs, manifests etc. You will need to run additional software to proceed.

Hosted

Delegated

Hosted -> RIPE se encarga de todo 😊

By clicking on 'accept' below you confirm that that you have read, understood and agree to the [RIPE NCC Certification Service Terms and Conditions](#).

RPKI RPKI dashboard

Documentation

Feedback/Support

Legal
Data, RPL, Privacy, Terms and Cookies

- My LIR
LIR Account, Billing, Users, General Meeting...
- Requests
Tickets, Resources, Updates, Transfers
- Resources
My Resources, Sponsored Resources
- RIPEDatabase
- RPKI
RPKI Dashboard

2 BGP Announcements
0 Valid 1 Invalid 2 Unknown

0 ROAs
0 OK 0 Causing problems

BGP Announcements [Route Origin Authorisations \(ROAs\)](#) [History](#)

Create ROAs for selected BGP Announcements Valid Invalid Unknown

<input type="checkbox"/> Origin AS	Prefix	Current Status	
<input type="checkbox"/> AS [Redacted]	185 [Redacted] /24	UNKNOWN	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> AS [Redacted]	185 [Redacted] /24	UNKNOWN	<input type="checkbox"/> <input type="checkbox"/>

Show 25

[Looking for ROA Certification for PI resources?](#)

[Revoke hosted CA](#)

- Documentation
- Feedback/Support
- Legal
Copyright, Privacy, Terms and Cookies



- My LIR
LIR Account, Billing, Users, General Meeting...
- Requests
Tickets, Resources, Updates, Transfers
- Resources
My Resources, Sponsored Resources
- RIPE Database
- RPKI
RPKI Dashboard

RPKI Dashboard

3 CERTIFIED RESOURCES NO ALERT EMAIL CONFIGURE

2 BGP Announcements

0 ROAs

0 Valid

0 Invalid

2 Unknown

0 OK

0 Causing problems

BGP Announcements		Route Origin Authorisations (ROAs)	History
<input checked="" type="checkbox"/> Create ROAs for selected BGP Announcements		<input type="checkbox"/> Valid <input type="checkbox"/> Invalid <input type="checkbox"/> Unknown	
Origin AS	Prefix	Current Status	
<input checked="" type="checkbox"/> AS: [redacted]	185. [redacted] /24	UNKNOWN	
<input checked="" type="checkbox"/> AS: [redacted]	185. [redacted] /24	UNKNOWN	

aplicamos cambios

Current Status UNKNOWN UNKNOWN

Seleccionamos los prefijos

- Documentation
- Feedback/Support
- Legal
Copyright, Privacy, Terms and Cookies

Revoke hosted CA

The RPKI dashboard will be unavailable on Thursday 19 August 2021 from 08:30-10:30 (UTC) due to planned maintenance. Please refer to our [service announcements](#) for more information.

RPKI Dashboard 3 CERTIFIED RESOURCES NO ALERT EMAIL CONFIGURE

2 BGP Announcements 0 ROAs

0 Valid 0 Invalid 2 Unknown 0 OK 0 Causing problems

BGP Announcements		Route Origin Authorisations (ROAs)	History
<input type="checkbox"/> Create ROAs for selected BGP Announcements		<input type="checkbox"/> Valid <input type="checkbox"/> Invalid <input type="checkbox"/> Unknown	
Origin AS	Prefix	Current Status	Future Status
<input type="checkbox"/> AS: [redacted]	185. [redacted].0/24	UNKNOWN	VALID
<input type="checkbox"/> AS: [redacted]	185. [redacted].0/24	UNKNOWN	VALID

Showing 25 items

[Looking for ROA Certification for PI resources?](#) [Revoke hosted CA](#)

Review and publish changes



You are editing [redacted]

- My LIR**
LIR Account, Billing, Users, General Meeting...
- Requests**
Tickets, Resources, Updates, Transfers
- Resources**
My Resources, Sponsored Resources
- RIPE Database**
- RPKI**
RPKI Dashboard

RPKI Dashboard

3 CERTIFIED RESOURCES NO ALERT EMAIL CONFIGURE

2 BGP Announcements

0 Valid 0 Invalid 2 Unknown

0 ROAs

0 OK 0 Causing problems

BGP Announcements [Route Origin Authorisations \(ROAs\)](#) [History](#)

Create ROAs for selected BGP Announcements Valid Invalid Unknown

Origin AS	Prefix	Current Status	Future Status	
<input type="checkbox"/> AS [redacted]	185.[redacted]/24	UNKNOWN	VALID	
<input type="checkbox"/> AS [redacted]	185.[redacted]/24	UNKNOWN	VALID	

Show 25

[Looking for ROA Certification for PI resources?](#)

[Revoke hosted CA](#)

- Documentation
- Feedback/Support
- Legal
Copyright, Privacy, Terms and Cookies

Review and publish changes ²

Staged ROAs

- AS [redacted] 185.[redacted]/24 → 24
- AS [redacted] 185.[redacted].0/24 → 24

Affected announcements

- AS [redacted] 185.[redacted]/24 UNKNOWN
- AS [redacted] 185.[redacted].0/24 UNKNOWN



- My LIR**
LIR Account, Billing, Users, General Meeting...
- Requests**
Tickets, Resources, Updates, Transfers
- Resources**
My Resources, Sponsored Resources
- RIPE Database**
- RPKI**
RPKI Dashboard

You are editing [redacted]

RPKI Dashboard 3 CERTIFIED RESOURCES NO ALERT EMAIL CONFIGURE

2 BGP Announcements

2 Valid 0 Invalid 0 Unknown

2 ROAs

2 OK

0 Causing problems

BGP Announcements Route Origin Authorisations (ROAs) History

↓ Create ROAs for selected BGP Announcements

Origin AS	Prefix	Current Status
<input type="checkbox"/> AS: [redacted]	185. [redacted] /24	VALID
<input type="checkbox"/> AS: [redacted]	185. [redacted] /24	VALID

Show 25

configurar alertas al correo



Looking for ROA Certification for PI resources? Revoke hosted CA

- Documentation
- Feedback/Support
- Legal
Copyright, Privacy, Terms and Cookies

Una vez firmados ya se puede ir a tomar café con su nuevo amigo:



Generación de Prefix List automática



como puedo obtener todos los prefijos de un AS



Para obtener todos los prefijos anunciados por un Sistema Autónomo (AS), puedes utilizar herramientas y bases de datos de enrutamiento BGP. A continuación, te presento dos métodos comunes:

1. Utilizando la base de datos de enrutamiento BGP (como RIPEstat):

- Accede al sitio web de RIPEstat (<https://stat.ripe.net/>) u otra base de datos de enrutamiento BGP.
- Ingresa el número de AS del que deseas obtener los prefijos en el campo de búsqueda.
- Busca la opción o pestaña relacionada con la información de enrutamiento BGP para ese AS.
- La base de datos mostrará una lista de prefijos anunciados por ese AS, junto con otra información relevante, como la longitud del prefijo y el origen del anuncio.
- Puedes exportar la información o copiarla en un archivo para su uso posterior.

2. Utilizando herramientas de línea de comandos o scripts:

- Utiliza herramientas como "bgp" en sistemas operativos Unix o "show ip bgp" en enrutadores Cisco.
- Conecta a un enrutador BGP y ejecuta el comando adecuado, especificando el número de AS del que deseas obtener los prefijos.
- La salida mostrará la tabla de enrutamiento BGP del AS especificado, que incluirá todos los prefijos anunciados por ese AS.
- Puedes redirigir la salida a un archivo para guardarla o utilizar herramientas de scripting para procesarla según sea necesario.

Recuerda que los resultados pueden variar según la fuente de datos utilizada y las políticas de visualización de información de cada proveedor. Además, ten en cuenta que algunos proveedores pueden requerir autenticación para acceder a su base de datos de enrutamiento BGP o enrutadores.



como puedo obtener todos los prefijos de un AS y generar un prefix-list automáticamente



Para obtener todos los prefijos de un Sistema Autónomo (AS) y generar una lista de prefijos automáticamente, puedes utilizar herramientas y scripts para extraer la información de la tabla de enrutamiento BGP y luego generar la lista de prefijos. A continuación, te presento un ejemplo de cómo podrías hacerlo utilizando Python y la librería 'pytricia':

1. Instala la librería 'pytricia' utilizando el siguiente comando en la terminal:

```
pip install pytricia
```

2. Crea un script de Python con el siguiente código:

```
python
from pytricia import PyTricia
from subprocess import check_output

# Obtén la tabla de enrutamiento BGP para el AS y guárdala en un archivo
as_number = "12345" # Reemplaza con el número de AS deseado
output_file = "bgp_table.txt"
command = f"show ip bgp {as_number} > {output_file}" # Comando específico de
check_output(command, shell=True)

# Lee el archivo de la tabla de enrutamiento y extrae los prefijos
prefix_list = PyTricia()
with open(output_file, "r") as file:
    for line in file:
        line = line.strip()
        if line.startswith("BGP routing table entry"):
            parts = line.split()
            prefix = parts[0]
            prefix_list[prefix] = True

# Genera el archivo prefix-list con los prefijos extraídos
prefix_list_file = "prefix_list.txt"
with open(prefix_list_file, "w") as file:
    for prefix in prefix_list.keys():
        file.write(f"ip prefix-list PREFIX_LIST permit {prefix}\n")
```

3. Reemplaza 'as_number' con el número de AS del que deseas obtener los prefijos.
4. Ejecuta el script de Python y generará un archivo llamado 'prefix_list.txt' que contendrá una lista de comandos 'ip prefix-list' para permitir todos los prefijos extraídos de la tabla de enrutamiento BGP.

Puedes ajustar el código según tus necesidades específicas, como la forma en que se ejecuta el comando 'show ip bgp' en tu enrutador o la estructura de la lista de prefijos generada.

Recuerda que el proceso puede variar según el enrutador y la versión del sistema operativo que estás utilizando. Asegúrate de consultar la documentación específica de tu enrutador y adaptar el código en consecuencia.

Generación de Prefix List automática

no hay otro metodo más simple ?

Si estás buscando un método más simple para obtener todos los prefijos de un Sistema Autónomo (AS) y generar una lista de prefijos automáticamente, puedes considerar el uso de herramientas de automatización de red como NAPALM o Netmiko. Estas herramientas te permiten interactuar con enrutadores y conmutadores de red mediante scripts de Python. Aquí hay un ejemplo utilizando la biblioteca NAPALM:

1. Instala la biblioteca NAPALM utilizando el siguiente comando en la terminal:

```
pip install napalm
```

2. Crea un script de Python con el siguiente código:

```
python
from napalm import get_network_driver

# Configuración del dispositivo
driver = get_network_driver('ios') # Reemplaza 'ios' por el nombre del sistema
device_ip = '192.168.1.1' # Reemplaza con la dirección IP del enrutador
username = 'username' # Reemplaza con el nombre de usuario
password = 'password' # Reemplaza con la contraseña

# Conexión al enrutador
device = driver(device_ip, username, password)
device.open()

# Obtención de la tabla de enrutamiento BGP
bgp_table = device.get_bgp_neighbors()

# Generación de la lista de prefijos
prefix_list = []
for neighbor, data in bgp_table.items():
    for prefix in data['address_family']['ipv4']['received_prefixes']:
        prefix_list.append(prefix)

# Impresión de la lista de prefijos
for prefix in prefix_list:
    print(prefix)

# Cierre de la conexión
device.close()
```

Método Carlos (ej. con AS204773 - DMNTR Networks)

1) Whois: `carlos@confsrv01:~$ whois -h whois.ripe.net -- '-a -r -i or -T route AS204773' | grep route: | awk '{print $2}' | sort -V`

185.66.75.0/24

2) BGPQ4: `carlos@confsrv01:~$ bgpq4 -I DMNTR AS204773`

no ip prefix-list DMNTR
ip prefix-list DMNTR permit 185.66.75.0/24

* <https://github.com/bgp/bgpq4>

RPKI - VALIDACIÓN

Algunas guías en Internet:

https://bgpfilterguide.nlnog.net/guides/reject_invalids/

<https://afenieux.fr/blog/article/dropping-rpki-roa-invalid-routes-on-arista-eos>

https://labs.ripe.net/documents/64/BGP_RPKI_Tofoni_Luciani_20200319.pdf

<https://beufa.net/blog/rpki-use-routinator-rtr-cache-validator-cisco-ios-xr/>



me puedes generar una configuración para Arista para incluir un validador de prefijos RPKI

Por supuesto, aquí tienes un ejemplo de configuración para Arista que incluye la integración de un validador de prefijos RPKI:

1. Accede al modo de configuración de Arista:

```
bash  Copy code
enable
configure terminal
```

2. Configura la conexión al servidor RPKI:

```
php  Copy code
ip rtr-rpki server <servidor_RPKI> vrf <vrf_name>
```

Donde **<servidor_RPKI>** es la dirección IP o el nombre de dominio del servidor RPKI y **<vrf_name>** es el nombre de la instancia de enrutamiento virtual (VRF) en caso de que se esté utilizando.

3. Habilita la validación de prefijos en una interfaz:

```
cssharp  Copy code
interface <interface_name>
ip rtr-rpki origin-validation
```

Cada uno es soberano en su red, ¿por qué una vrf ?

Necesidad de aporte de más información de la estructura de red ...
confidencialidad (Italia prohíbe el uso de chatGPT)

<https://elpais.com/tecnologia/2023-03-31/italia-bloquea-el-uso-de-chatgpt-por-incumplir-la-normativa-de-proteccion-de-datos.html>

Instalamos ROUTINATOR, en docker (así GruBuP aprende algo nuevo)



podemos tener el software routinator en docker? me generas el archivo para docker compose



Sí, Routinator se puede ejecutar en un contenedor Docker. Aquí tienes un ejemplo de archivo `docker-compose.yml` para ejecutar Routinator en Docker:

```
yaml Copy code
version: '3'

services:
  routinator:
    image: nlnetlabs/routinator:latest
    ports:
      - 3323:3323
    volumes:
      - ./routinator.conf:/home/routinator/.rpki-cache/routinator.conf
      - ./data:/home/routinator/.rpki-cache
    restart: always
```

Asegúrate de seguir estos pasos:

1. Crea un archivo llamado `docker-compose.yml`.
2. Copia y pega el contenido del ejemplo anterior en el archivo `docker-compose.yml`.
3. Crea un archivo de configuración de Routinator llamado `routinator.conf` y colócalo en el mismo directorio que el archivo `docker-compose.yml`. Asegúrate de ajustar los valores de configuración según tus necesidades.
4. Crea una carpeta llamada `data` en el mismo directorio que el archivo `docker-compose.yml`. Esta carpeta se utilizará para almacenar los datos y caché de Routinator.
5. Ejecuta el siguiente comando en la misma ubicación que el archivo `docker-compose.yml`:



```
docker-compose up -d
```

Esto iniciará el contenedor de Routinator en segundo plano.

Ahora tendrás Routinator ejecutándose en un contenedor Docker. El contenedor expondrá el puerto 3323, que puedes mapear al puerto de tu elección en la configuración del servicio en el archivo `docker-compose.yml`.

Recuerda adaptar la configuración y las rutas de los archivos según tus necesidades. Además, asegúrate de consultar la documentación oficial de Routinator para obtener más información sobre la configuración y el uso de Routinator en un contenedor Docker.

hay que fijarse en el path, pero se ejecuta sin problema alguno:

```
carlos@confsrv01:~/docker$ sudo docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                                     NAMES
2bfbfd8be855  nlnetlabs/routinator:latest        "/sbin/tini -- routi..." 48 seconds ago Up 7 seconds  8323/tcp, 0.0.0.0:3323->3323/tcp, :::3323->3323/tcp, 9556/tcp  docker_routinator_1
```

No funciona por algún motivo, ya os digo que es mi culpa 🙄

```
Connection: Connect (Connection refused)
Transport information:
Protocol: TCP
```

Lo instalo sin docker pero sigue sin funcionar 😭

CA intalar routinator en un sistema GNU/Linux ubuntu

Para instalar Routinator en un sistema GNU/Linux Ubuntu, puedes seguir estos pasos:

1. Actualiza los repositorios del sistema:

```
sql
sudo apt update
```
2. Instala las dependencias necesarias:

```
arduino
sudo apt install -y build-essential libssl-dev libpq-dev pkg-config
```
3. Instala Rust utilizando rustup:

```
arduino
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```

Sigue las instrucciones en pantalla y selecciona la opción predeterminada para instalar Rust.
4. Agrega Rust al PATH ejecutando el siguiente comando:

```
bash
source $HOME/.cargo/env
```

5. Descarga Routinator:

```
bash
git clone https://github.com/NlnetLabs/routinator.git
```

6. Navega hasta el directorio de Routinator:

```
bash
cd routinator
```

7. Compila e instala Routinator:

```
css
cargo build --release
cargo install --path .
```

Esto compilará y construirá Routinator en tu sistema.

Una vez que hayas seguido estos pasos, Routinator estará instalado en tu sistema GNU/Linux Ubuntu. Puedes ejecutarlo utilizando el comando `routinator`. Asegúrate de consultar la documentación oficial de Routinator para obtener más información sobre la configuración y el uso de la herramienta.

Recuerda que estos pasos son una guía general y pueden variar según tu entorno específico. Si encuentras algún problema durante la instalación, consulta la documentación oficial de Routinator o busca ayuda en la comunidad de Routinator.

CA como puedo acabar de configurar routinator en mi sistema GNU/Linux

Después de instalar Routinator en tu sistema GNU/Linux, aquí tienes los pasos adicionales para completar la configuración:

1. Generar una configuración inicial:

Routinator requiere una configuración inicial para funcionar correctamente. Puedes generar una configuración básica utilizando el siguiente comando:

```
cahierp
routinator init --accept-arin-rpa
```

Esto generará un archivo de configuración llamado `routinator.conf` en el directorio actual y aceptará automáticamente los términos del acuerdo de uso de datos de ARIN.
2. Editar la configuración de Routinator:

Abre el archivo `routinator.conf` en un editor de texto y ajusta los valores según tus necesidades. Algunas configuraciones comunes que puedes ajustar incluyen:

 - `repository_rdp`: Especifica la URL del repositorio RPKI que deseas utilizar. Puedes encontrar las URLs de los repositorios en los sitios web de las organizaciones RIR (Regional Internet Registry) correspondientes.
 - `listen`: Configura la dirección IP y el puerto en el que Routinator escuchará las conexiones entrantes para las actualizaciones RPKI.
3. Descargar los datos RPKI:

```
sql
routinator update
```

Este comando descargará los datos RPKI del repositorio especificado en la configuración.

Al final he acabado en la página del proyecto <https://routinator.docs.nlnetlabs.nl/en/stable/index.html>

Como GruBuP no lo ha conseguido ... se queda sin café

El "RPKI Team" se van a tomar café sin él



Gestión del Peering - IX-Point

IX-Points España:

- ESPANIX (Madrid) - 1997
- CATNIX (Barcelona) - 1999
- NIXVAL (Valencia) - 2016
- DE-CIX (Bcn-Mad-Vlc) - 2016

 me puedes decir todos los IX-POINTS que dispone ESPAÑA, ordena según año de creación

 A continuación te presento una lista de los principales puntos de intercambio de Internet (IX-Points) en España, ordenados por año de creación:

1. CATNIX (Catalunya Neutral Internet Exchange)
 - Año de creación: 1999
 - Ubicación: Barcelona
 - Sitio web: <https://www.catnix.net/>
2. Espanix (Spain Internet Exchange)
 - Año de creación: 2000
 - Ubicación: Madrid
 - Sitio web: <https://www.espanix.net/>
3. DE-CIX Madrid (German Commercial Internet Exchange)
 - Año de creación: 2016
 - Ubicación: Madrid
 - Sitio web: <https://www.de-cix.net/es/locations/madrid>
4. LINX NoVA (London Internet Exchange Northern Virginia)
 - Año de creación: 2017 (expansión a España)
 - Ubicación: Barcelona
 - Sitio web: <https://www.linx.net/services/nova/>
5. BCN-IX (Barcelona Internet Exchange)
 - Año de creación: 2018
 - Ubicación: Barcelona
 - Sitio web: <https://www.bcn-ix.net/>
6. ESPANIX-MADRID (ESPAÑA)
 - Año de creación: 2021
 - Ubicación: Madrid
 - Sitio web: <https://www.espanix.net/>

Estos son algunos de los principales IX-Points en España, ordenados por año de creación. Ten en cuenta que pueden existir otros puntos de intercambio más pequeños o regionales en diferentes ciudades de España. Te recomendaría visitar los sitios web de cada IX-Point para obtener información actualizada sobre su ubicación, miembros y servicios ofrecidos.

Gestión del Peering - IX-Point

EA

que herramientas me recomiendas para tener el peering con un IX-Point automatizado



Existen varias herramientas que puedes utilizar para automatizar el proceso de peering con un IX-Point (Punto de Intercambio de Internet). A continuación, te mencionaré algunas opciones populares:

1. **IXP Manager:** Es una plataforma de gestión de Puntos de Intercambio de Internet de código abierto. Proporciona funciones para administrar de forma automatizada el proceso de peering, incluyendo la configuración de sesiones BGP, el monitoreo y la generación de rutas de filtrado. También facilita la gestión de miembros, el registro de recursos y la visualización de estadísticas.
2. **PeeringDB:** No es una herramienta de automatización en sí, pero es una base de datos pública utilizada por operadores de red y proveedores de servicios para intercambiar información sobre peering. Puedes utilizar la API de PeeringDB para acceder a los datos y automatizar la configuración de sesiones BGP y la gestión de peering.
3. **Ansible:** Es una plataforma de automatización de TI de código abierto que permite la configuración y el despliegue automatizado de dispositivos de red. Puedes utilizar Ansible para definir playbooks que automatizan la configuración de sesiones BGP, así como otras tareas relacionadas con el peering.
4. **NAPALM:** Es una biblioteca y conjunto de herramientas que proporciona una interfaz unificada para interactuar con dispositivos de red. NAPALM te permite automatizar tareas de configuración, incluyendo la configuración de sesiones BGP en los routers, a través de una API sencilla.
5. **SDN Controllers:** Si estás utilizando una arquitectura de red definida por software (SDN), puedes aprovechar los controladores SDN para automatizar el proceso de peering. Controladores populares como OpenDaylight o ONOS permiten la gestión automatizada de sesiones BGP y la configuración de rutas en entornos SDN.

Estas son solo algunas de las opciones disponibles. La elección de la herramienta depende de tus necesidades específicas, el entorno de red y la infraestructura existente. Te recomendaría evaluar cada opción y determinar cuál se ajusta mejor a tus requerimientos y habilidades técnicas.

EA

puedes facilitarme otras opciones



¡Por supuesto! Aquí tienes algunas opciones adicionales para automatizar el peering con un IX-Point:

1. **ExaBGP:** Es un software de enrutamiento BGP de código abierto que se ejecuta en sistemas Linux y permite automatizar el establecimiento y la gestión de sesiones BGP. Puedes utilizar ExaBGP para configurar anuncios y filtros de ruta de manera programática.
2. **Python BGP Library (PyBGP):** Es una biblioteca de Python que facilita la implementación de funcionalidades BGP, incluyendo la configuración y el control de sesiones BGP. PyBGP te permite desarrollar scripts personalizados para automatizar el proceso de peering.
3. **SaltStack:** Es una plataforma de automatización de TI y orquestación que puede ser utilizada para automatizar la configuración de sesiones BGP en dispositivos de red. SaltStack proporciona un enfoque basado en estados para la configuración y gestión de la infraestructura de red.
4. **Network Configuration Protocol (NETCONF) / YANG:** NETCONF es un protocolo de gestión de configuración de red basado en XML, y YANG es un lenguaje de modelado utilizado para describir la configuración de dispositivos de red. Puedes utilizar NETCONF y YANG para automatizar la configuración de sesiones BGP en dispositivos compatibles.
5. **Cloud-based Network Automation Platforms:** Existen plataformas basadas en la nube que ofrecen capacidades de automatización de red, incluyendo el peering con IX-Points. Algunas opciones populares incluyen Cisco DNA Center, Juniper Contrail, y Arista CloudVision, que proporcionan funcionalidades de automatización y gestión centralizada.

Recuerda evaluar cada opción en función de tus necesidades y requerimientos específicos. Además, ten en cuenta la compatibilidad de las herramientas con tu infraestructura y la curva de aprendizaje asociada.

IXP Manager

<https://www.ixpmanager.org/>



Peering Manager

Potential Peers | Potential Bilateral Peers | Peers | Rejected / Ignored Peers

You currently do not exchange any routes in any way with the following members of the exchange over the highlighted - in red - protocol(s) and LAN(s) because:

- either you, they or both of you are not route server clients; and
- you do not have a bilateral (direct) peering session that we have detected with them.

Member	ASN	Policy	Peering LAN1	Peering LAN2	INEX Cork	
Amazon.com	16509	open	IPv4 IPv6			<input type="checkbox"/> Request Peering <input type="checkbox"/> Notes
Apple Distribution International	714	selective	IPv4 IPv6	IPv4 IPv6	IPv4 IPv6	<input type="checkbox"/> Request Peering <input type="checkbox"/> Notes
Apple Distribution International - AS6185	6185	selective			IPv4 IPv6	<input type="checkbox"/> Request Peering <input type="checkbox"/> Notes
BT AS5400	5400	selective	IPv4 IPv6			<input type="checkbox"/> Request Peering <input type="checkbox"/> Notes
Convergence Group	41811	open	IPv4 IPv6			<input type="checkbox"/> Request Peering <input type="checkbox"/> Notes
Linsight Networks	22822	selective	IPv4 IPv6			<input type="checkbox"/> Request Peering <input type="checkbox"/> Notes
LinkedIn	14413	open	IPv4 IPv6			<input type="checkbox"/> Request Peering <input type="checkbox"/> Notes
Servecentric	15612	open	IPv4			<input type="checkbox"/> Request Peering <input type="checkbox"/> Notes
Verizon Business	702	selective	IPv4 IPv6			<input type="checkbox"/> Request Peering <input type="checkbox"/> Notes
Virgin Media Ireland Ltd	6830	selective	IPv4 IPv6	IPv4 IPv6		<input type="checkbox"/> Request Peering <input type="checkbox"/> Notes
Zendesk	61186	open	IPv4			<input type="checkbox"/> Request Peering <input type="checkbox"/> Notes

<https://docs.ixpmanager.org/>

<https://docs.ixpmanager.org/features/peering-manager/>

Peering Manager

<https://peering-manager.net/>



Peering Manager Internet Peering Autonomous Systems Deployment gpmayor

Peering Sessions

AS1	AS2 Name	IP Address	Status	State	AS1 Label	AS2 Label	IP Version	Enabled
42	Packet Clearing House AS42	2001.78.04.92	Enabled	Established	AS1: 42	AS2: 42	4	Enabled
42	Packet Clearing House AS42	37.49.236.92	Enabled	Established	AS1: 42	AS2: 42	4	Enabled
2209	RENAULT	2001.78.04.19	Enabled	Established	AS1: 2209	AS2: 2209	4	Enabled
2209	RENAULT	37.49.236.19	Enabled	Established	AS1: 2209	AS2: 2209	4	Enabled
2484	FR-NIC-EN5-ALVCAST (AFNIC / NIC-France)	2001.78.04.20	Enabled	Established	AS1: 2484	AS2: 2484	4	Enabled
2484	FR-NIC-EN5-ALVCAST (AFNIC / NIC-France)	37.49.236.20	Enabled	Established	AS1: 2484	AS2: 2484	4	Enabled
2486	FR-NIC-EN5-LINEAS1 (AFNIC / NIC-France)	2001.78.04.21	Enabled	Established	AS1: 2486	AS2: 2486	4	Enabled
2486	FR-NIC-EN5-LINEAS1 (AFNIC / NIC-France)	37.49.236.21	Enabled	Established	AS1: 2486	AS2: 2486	4	Enabled
2506	Netfilix	2001.78.04.130	Enabled	Established	AS1: 2506	AS2: 2506	4	Enabled
2506	Netfilix	37.49.237.00	Enabled	Established	AS1: 2506	AS2: 2506	4	Enabled
3035	Packet Clearing House	2001.78.04.91	Enabled	Config	AS1: 3035	AS2: 3035	4	Enabled
3035	Packet Clearing House	37.49.236.91	Enabled	Config	AS1: 3035	AS2: 3035	4	Enabled
4601	Post Group Luxembourg	2001.78.04.90	Enabled	Established	AS1: 4601	AS2: 4601	4	Enabled
4601	Post Group Luxembourg	37.49.236.90	Enabled	Established	AS1: 4601	AS2: 4601	4	Enabled
4639	Huyfane Electric	2001.78.04.13	Enabled	Established	AS1: 4639	AS2: 4639	4	Enabled
4639	Huyfane Electric	37.49.236.13	Enabled	Established	AS1: 4639	AS2: 4639	4	Enabled
8075	Microsoft	2001.78.04.5	Enabled	Established	AS1: 8075	AS2: 8075	4	Enabled
8075	Microsoft	37.49.236.5	Enabled	Established	AS1: 8075	AS2: 8075	4	Enabled
8500	ISI Internet	2001.78.04.42	Enabled	Established	AS1: 8500	AS2: 8500	4	Enabled
8500	ISI Internet	37.49.236.42	Enabled	Established	AS1: 8500	AS2: 8500	4	Enabled

Showing 1-20 of 66

9/16/2021 10:05:01 CEST

https://ripe76.ripe.net/presentations/13-Peering_Manager_RIPE76_LT.pdf

Peering automatizado acabado ... más café



MANRS

BUENOS MODALES en Internet

<https://www.manrs.org/>

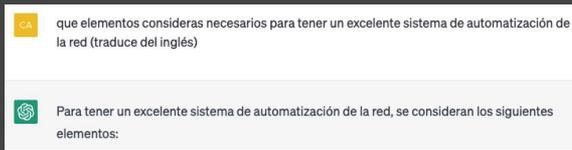
Prohibido perderse esta charla:

BEING A BETTER NETIZEN: MANRS
Mircea Ulinic (DigitalOcean)



Automatización:

¿Qué necesitamos ?



Gestión centralizada: Un sistema de automatización eficiente requiere una gestión centralizada de todos los dispositivos de red. Esto implica tener un repositorio central "source of truth" donde se almacene y administre la configuración de los dispositivos, así como una plataforma de gestión unificada desde donde se pueda supervisar y controlar la red.

Orquestación de red: La capacidad de orquestar las tareas de configuración y operación de la red es fundamental. Esto implica automatizar flujos de trabajo y procesos para simplificar y acelerar las tareas de aprovisionamiento, configuración y mantenimiento de los dispositivos de red.

Automatización de tareas repetitivas: Un buen sistema de automatización debe ser capaz de identificar y automatizar las tareas repetitivas, como la implementación de configuraciones estándar en múltiples dispositivos o la aplicación de actualizaciones de software de manera uniforme.

Telemetría y monitorización: La recopilación de datos y métricas en tiempo real es esencial para obtener una visibilidad completa de la red. Esto implica la implementación de tecnologías de telemetría y monitorización que permitan supervisar el rendimiento, la disponibilidad y la seguridad de la red.

Inteligencia y análisis de datos: La capacidad de recopilar, analizar y utilizar los datos de la red es clave para optimizar su rendimiento y tomar decisiones informadas. Los sistemas de automatización deben ser capaces de extraer información relevante de los datos recopilados y utilizarla para tomar acciones correctivas o mejorar la eficiencia operativa.

Seguridad integrada: Un sistema de automatización de la red debe tener una sólida capa de seguridad integrada. Esto implica implementar mecanismos de autenticación, autorización y auditoría, así como la capacidad de aplicar políticas de seguridad de manera coherente en todos los dispositivos de red.

Interoperabilidad y estándares abiertos: Para un sistema de automatización eficiente, es importante contar con dispositivos y soluciones que sean compatibles con estándares abiertos y puedan interoperar entre sí. Esto facilita la integración de diferentes componentes de la red y evita la dependencia de proveedores específicos.

Flexibilidad y escalabilidad: Un buen sistema de automatización debe ser flexible y escalable, capaz de adaptarse a los cambios en la red y crecer con las necesidades empresariales. Debe poder manejar tanto pequeñas redes como entornos empresariales complejos y en constante evolución.

Documentación y control de versiones: Es importante contar con una documentación actualizada y un control de versiones de las configuraciones de red. Esto permite realizar un seguimiento de los cambios realizados en la red, facilitar la resolución de problemas y revertir cambios si es necesario.

Colaboración y gestión del cambio: Un sistema de automatización exitoso fomenta la colaboración entre los equipos de redes, operaciones y desarrollo, y facilita la gestión efectiva del cambio. Esto implica implementar procesos y herramientas que permitan una comunicación fluida y una coordinación adecuada entre los diferentes actores involucrados.

Otros RECURSOS - hay que seguir formándose

RIPE NCC: <https://academy.ripe.net/>

LACNIC: <https://campus.lacnic.net/>

APNIC: <https://academy.apnic.net/en>

MANRS: <https://www.manrs.org/>

AUTOMATIZACION:

<https://wiki.geant.org/display/NETDEV/OAV+Training+Portal>



CONCLUSIONES:

- **ChatGPT nos puede de ser de gran ayuda**
- **ChatGPT no lo sabe todo** (al menos la versión gratuita, y miente con tanta seguridad ...)
- **Otros sistemas de IA en camino** (estaremos atentos ...)
- **No nos quita de leer y seguir estudiando.**
- **Es importante contrastar otras fuentes.**
- **Hay que tener capacidad crítica de análisis.**
- **Hay que saber que estamos haciendo y dónde queremos llegar en el proyecto o quizás se nos abre un mundo de posibilidades.**
- **Laboratorio siempre que sea posible**



Recursos presentación

Muñeco GruBuP: Lanukas - <https://lanukas.blogspot.com/>

Imagen portada: Image by [onlyyouqj](#) on Freepik
(https://www.freepik.com/free-photo/cup-coffee-with-pile-coffee-beans_903411.htm#query=coffee&position=7&from_view=search&track=sph)

Diagramas: Excalidraw <https://excalidraw.com/>
Network topology icons: [@dwelle](#) [@dwelle](#)

Lab Software: Eve-ng community edition <https://www.eve-ng.net/index.php/community/>

¿ Preguntas ?



Carlos Fraga



carlos@ipv6.cat

 <https://www.linkedin.com/in/cfragalo/>

 <https://github.com/cfragalo/ESNOG>