

<Code is the new CLI />

Network Programmability and beyond

Alfonso Sandoval Rosas
Software Consulting Engineer

WITCOM 2024

Acerca de

SWAT (Software & Automation Team) Lisboa **PT**

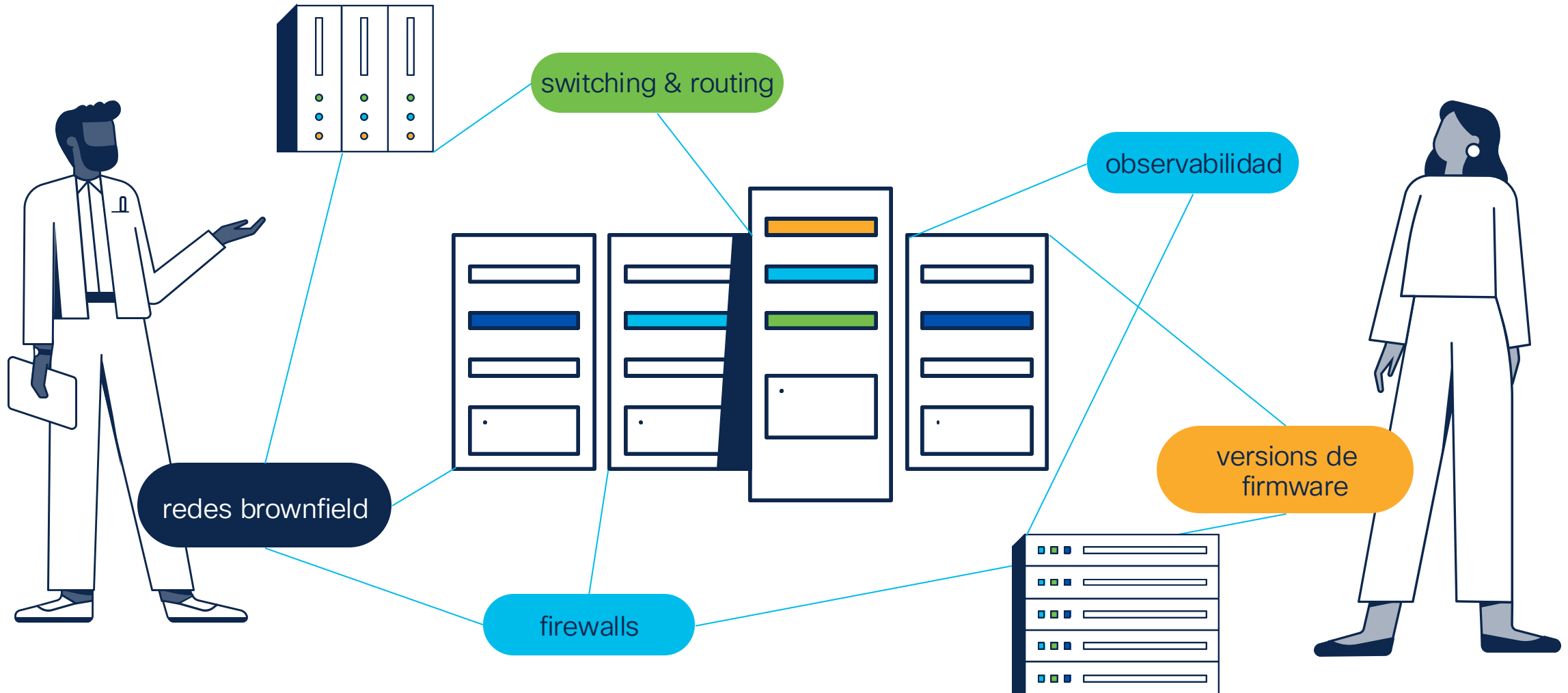
 github.com/ponchotitlan

 linkedin.com/in/asandovalros



Oficinas de Cisco en Tokio JP

POV: Administras una red de datos de gran escala (durante COVID-19)



OK, es imposible hacer esto vía SSH...

```
hostname XR-Router
domain name example.com
interface GigabitEthernet0/0/0/0
  description Link to core switch
  ipv4 address 192.0.2.1 255.255.255.0
  ipv6 address 2001:db8::1/64
  no shutdown
!
interface GigabitEthernet0/0/0/1
  description Link to core switch
  ipv4 address 198.51.100.1
  255.255.255.252
  ipv6 address 2001:db8:1::1/64
  no shutdown
router ospf 1
  router-id 1.1.1.1
  network 192.0.2.0 0.0.0.255 area 0
  passive-interface GigabitEthernet0/0/0/0
router bgp 65000
  bgp log-neighbor-changes
  neighbor 198.51.100.2 remote-as 65001
  neighbor 198.51.100.2 description ISP
  Connection
```



Múltiples fuentes de información



Copiar/pegar manualmente



Tiempo desperdiciado en
tipos y otros errores manuales

“

Ah, claro. Conozco a alguien quien tiene los dispositivos, vendors y configuraciones **en una hoja de cálculo ...(...)**



¿Qué suele salir mal?



Configuraciones masivas

Consistencia y atomicidad

Monitoreo entre dominios

Integración con el negocio

Por supuesto que creamos un script. Sin embargo ...



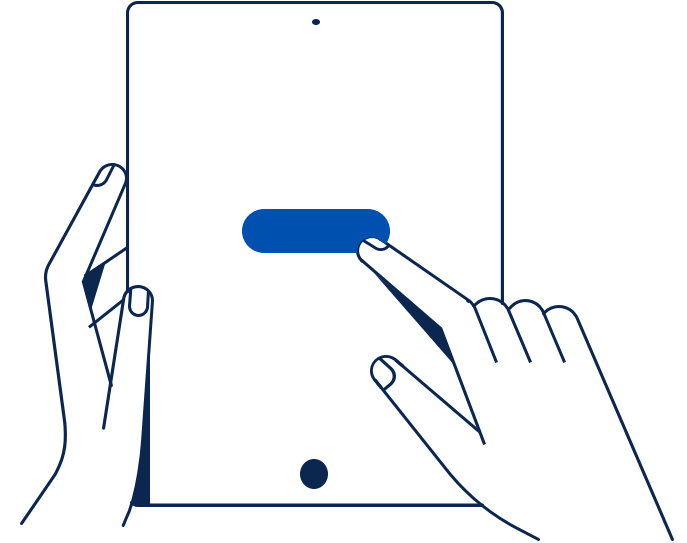
Simulación de comandos vía CLI



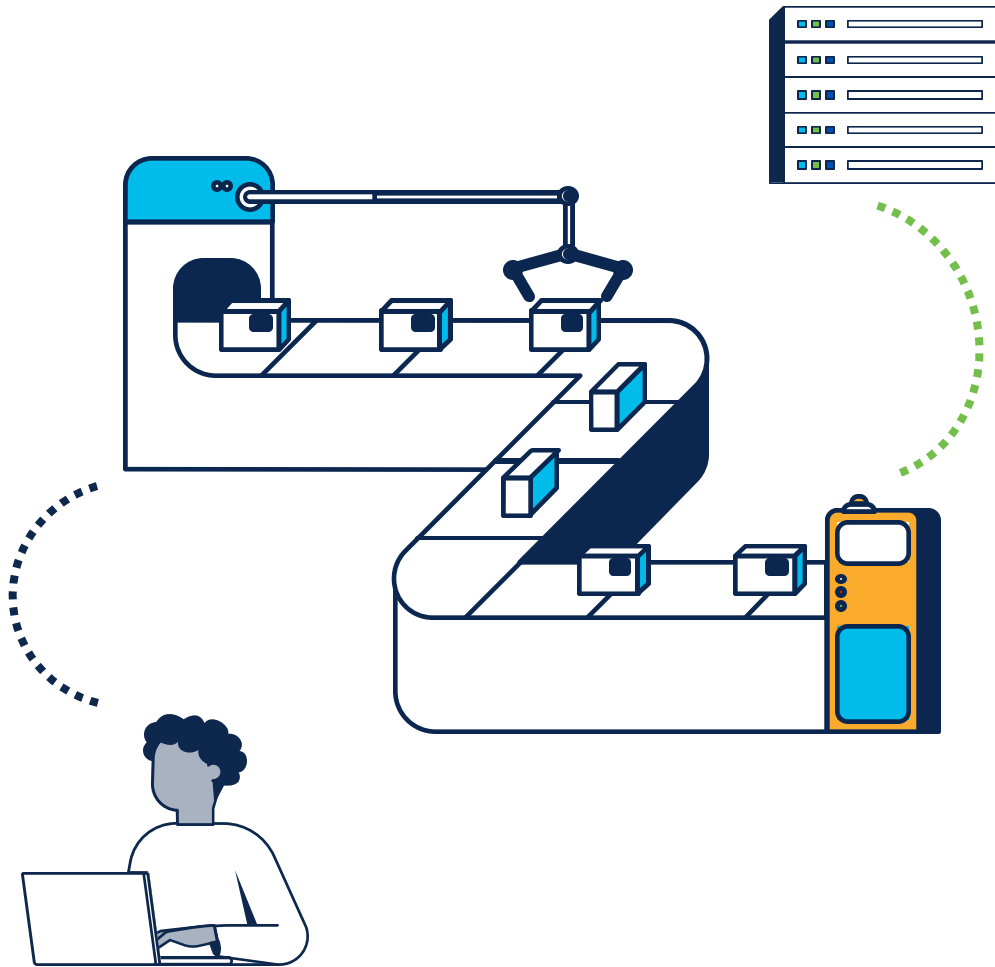
Sin control de commits (configs inválidas, healthchecks, etc)



Los comandos de CLI cambian entre versiones



La Interfaz CLI está hecha para ser usada por humanos. **No por máquinas**



Definir mi red con simple código fuente

Usar modelos estandarizados para interactuar con mi red

Automatizar los servicios en mi red

Integrar mi red con otros sistemas

Agenda



Programabilidad basada en modelos



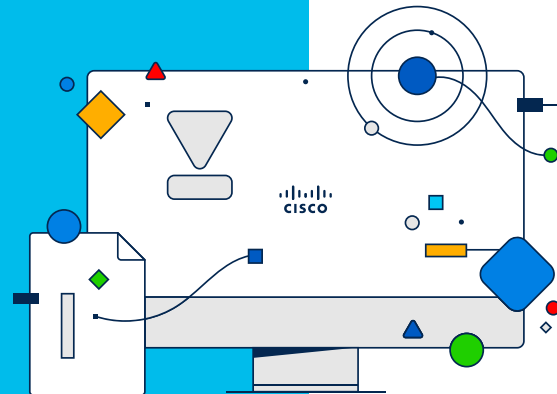
Protocolos NETCONF y RESTCONF



IaC (Infraestructura como Código)



Tips de carrera profesional



Programabilidad basada en modelos



Vía consola SSH/Telnet

Estructuras variables por cada vendor

Difícil rastrear los cambios entre versiones

Impráctico integrar con otros sistemas



Vía payloads basados en modelos

Estructuras estandarizadas (RFCs)

Posible versionar cambios en git al ser archivos de texto

Integración con otros sistemas



Modelado de datos con yang

- RFC 6020
- Lenguaje de modelado de datos
- Modelan las configuraciones y datos de estado de un dispositivo o servicio
- Organizados en nodos con tipos de datos
- Estándares industriales vs. Específicos del vendor
- Modelos de dispositivos (Interface, VLAN, etc)
- Modelos de servicios (L3VPN, VRF, etc)

arista-acl.yang

huawei-bgp.yang

cisco-xr-alarms.yang

juniper-interfaces.yang

cisco-xr-vlan.yang



Modelado de datos con yang

Leaf



```
leaf hostname {  
    type string;  
    description "Nombre del dispositivo"  
}
```

leaf representa el tipo de dato más pequeño, con solo un valor y un tipo de dato

Modelado de datos con yang

Leaf-list

```
leaf-list dns-servers {  
  type string;  
  description "Lista de direcciones IP de servidores DNS";  
}
```

leaf-list representa una colección ordenada de elementos leaf

Modelado de datos con yang

List 

```
list interface {  
  key "name";  
  leaf name {  
    type string;  
    description "Nombre de la interfaz";  
  }  
  leaf mtu {  
    type uint32;  
    description "Tamaño de MTUs de la interfaz";  
  }  
}
```

El tipo de dato **List** representa varios elementos, cada uno con multiples sub-elementos en un formato llave-valor

Modelado de datos con yang

Container

```
container system {  
  leaf hostname {  
    type string;  
    description "Nombre de host del dispositivo";  
  }  
  leaf timezone {  
    type string;  
    description "Zona horaria del dispositivo";  
  }  
}
```

Un **container** agrupa elementos que están relacionados a nivel semántico

Protocolos NETCONF y RESTCONF

Protocolo NETCONF

- RFC 6241
- Commit, edición y eliminación de configuraciones
- Sucesor de SNMP, pero para config y monitoreo
- Formato XML
- Comandos via RPCs (Remote Procedure Calls)
- Payload completamente basado en modelos YANG

1. Sesión SSH y mensaje “hello” para descargar la lista de modelos de datos YANG que el dispositivo soporta

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability> ... :base:1.0</capability>
    <capability> ... :base:1.1</capability>
    <capability> ... :writable-running:1.0</capability>
    <capability> ... :xpath:1.0</capability> ...
```

2. RPC request / RPC reply

```
<?xml version="1.0" encoding="utf-8"?>
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
<access-control xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-ipv4-acl-cfg">
  <accesses>
    <access>
      <access-list-name>ACL-EXAMPLE</access-list-name>
      <access-list-entries>
        <access-list-entry>
          <sequence-number>10</sequence-number>
          <grant>permit</grant>
          <source-network>
            <source-address>192.168.1.0</source-address>
            <source-wild-card-bits>0.0.0.255</source-wild-card-bits>
          </source-network>
        </access-list-entry>
        ...
      </access-list-entries>
    </access>
  </accesses>
</access-control>
</config>
</edit-config>
</rpc>
```

operaciones

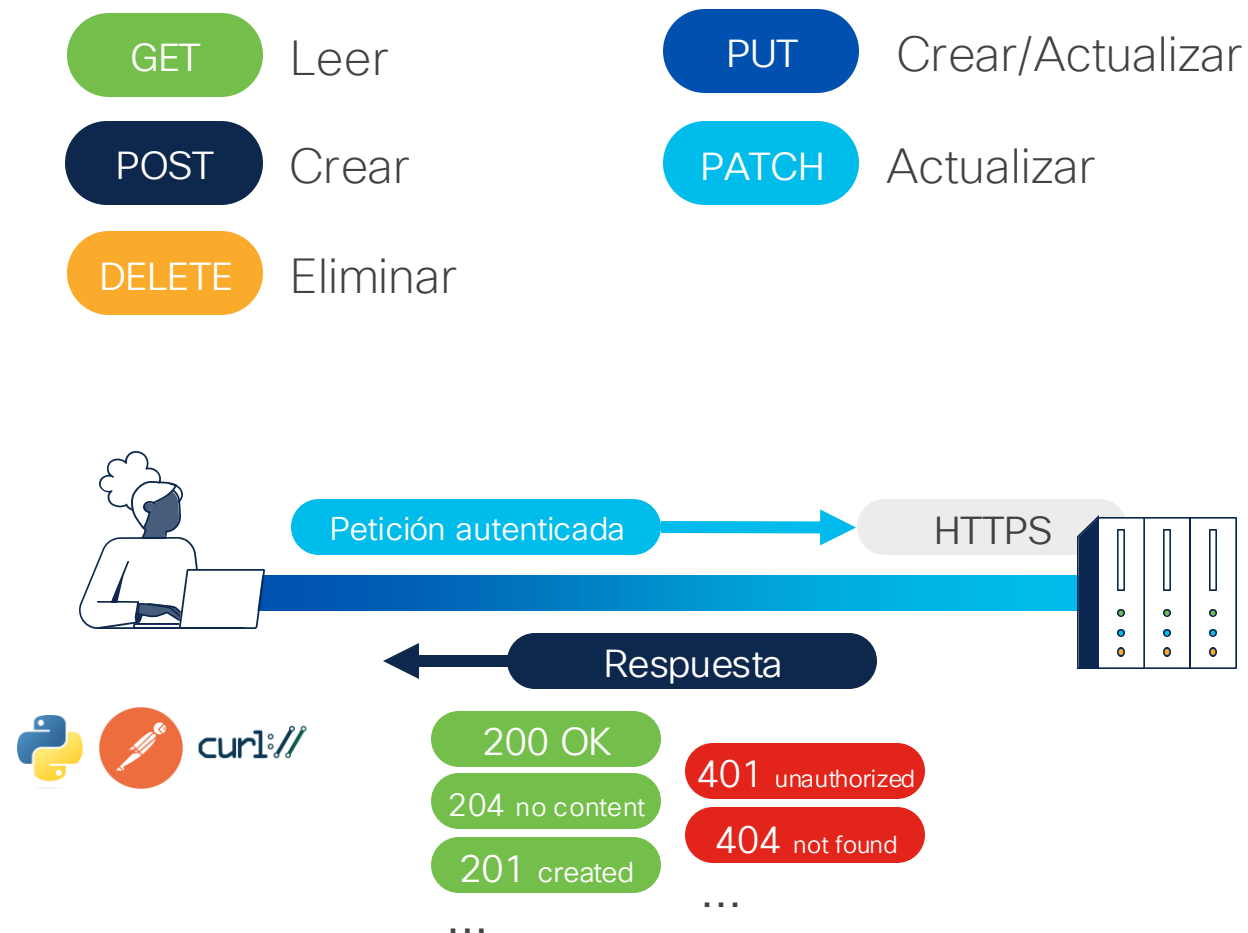
mensaje

payload basado en un modelo de datos



¿Recuerdas REST?

- Comunicación basada em HTTPS
- Stateless
- Interfaces RESTful API
- Verbos para acciones CRUD
- Códigos de respuesta estandarizados



Protocolo RESTCONF

- RFC 8040
- Formato XML o JSON
- Interacción RESTful con dispositivos
- GET, POST, PUT, PATCH, DELETE
- Transporte HTTPS
- También basado en modelos YANG

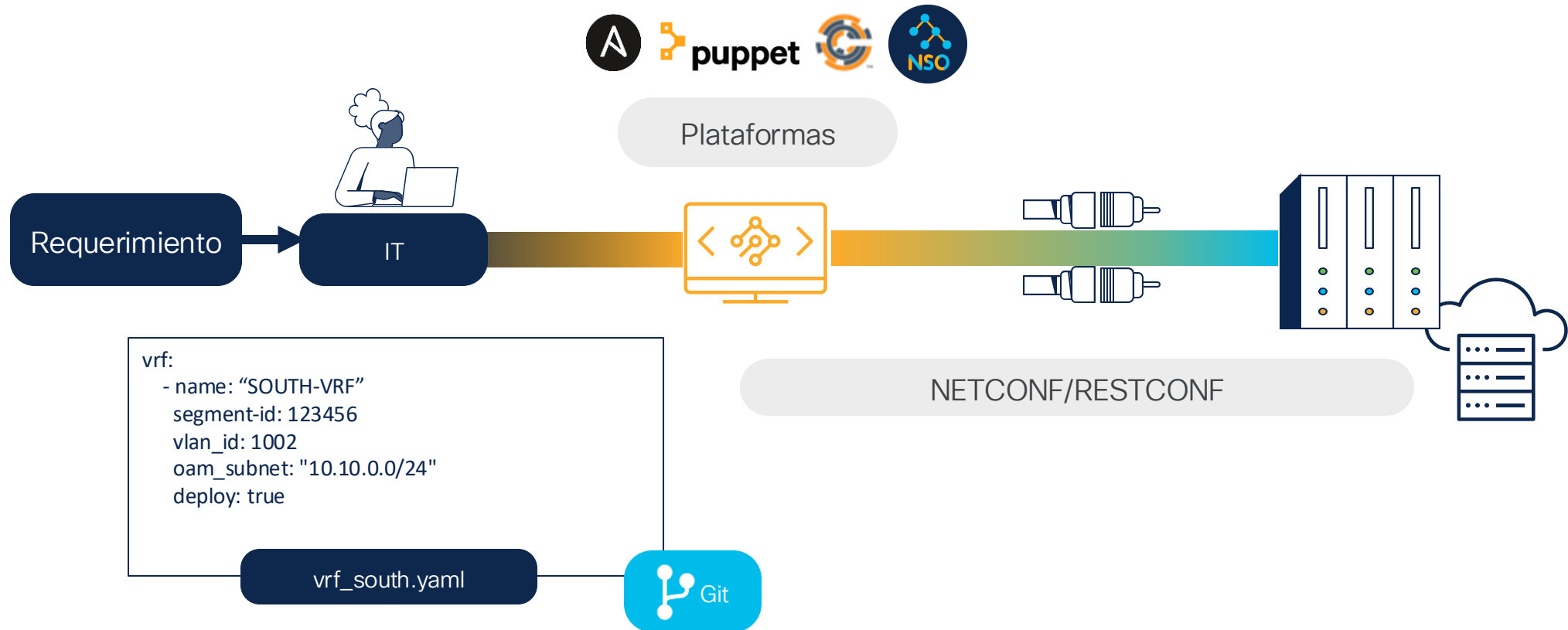
```
curl -k -u "your_username:your_password" \  
-H "Accept: application/yang-data+json" \  
-H "Content-Type: application/yang-data+json" \  
-X GET \  
"https://<device_ip>/restconf/data/ietf-  
interfaces:interfaces?depth=unbounded"
```



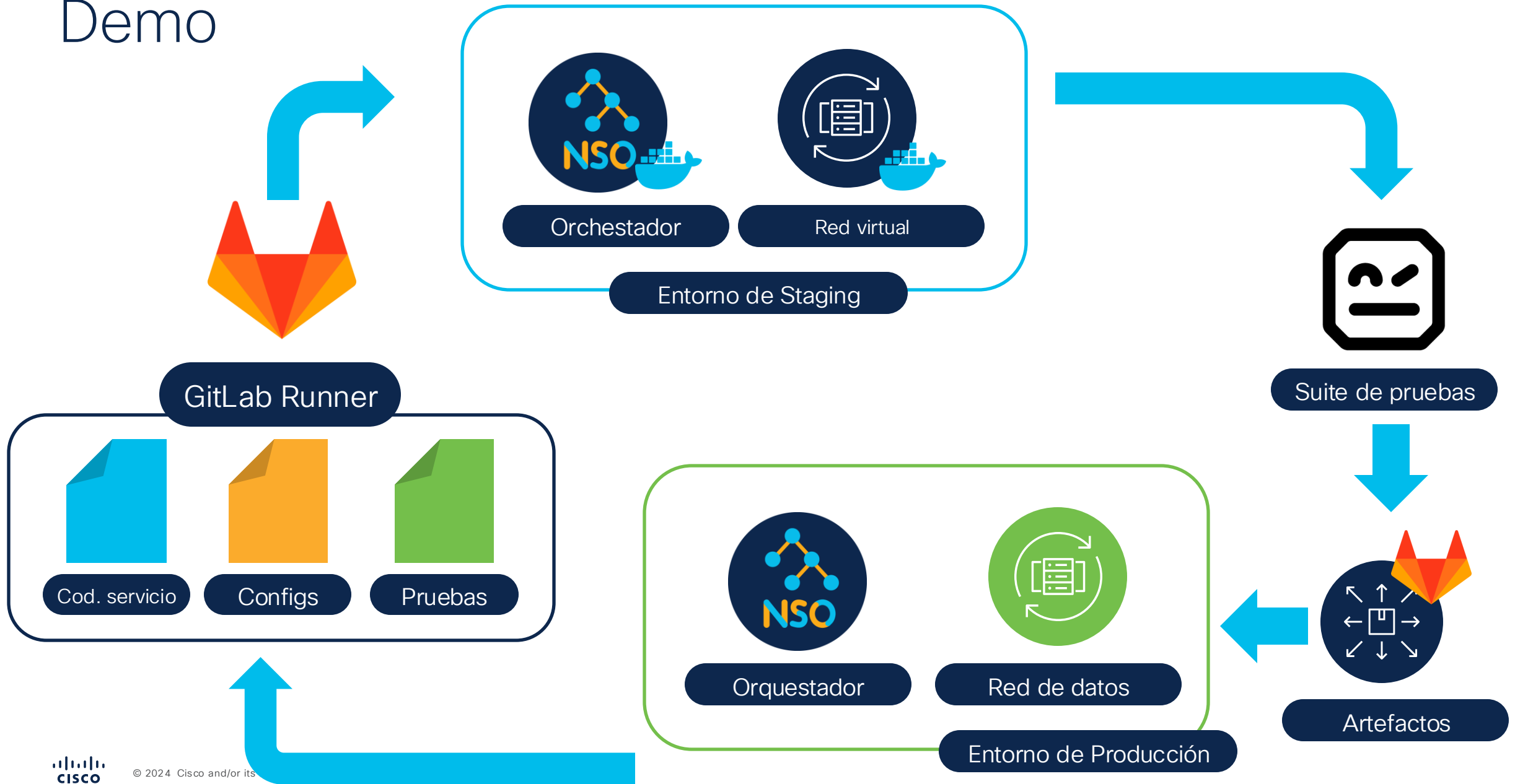
IaC (Infraestructura como Código)

IaC (Infrastructure as Code)

- Definición del estado deseado de nuestra red de datos mediante archivos de texto
- Diferentes plataformas de IaC para varios propósitos
- Plugins para diversos dispositivos



Demo



Tips de carrera profesional



Ingeniería de Software

- Patrones de diseño
- Programación Orientada a Objetos
- Documentación
- Git
- Unit testing
- Python suele ser el lenguaje más usado en el mundo de las redes



Ingeniería de Redes

- Conceptos de Routing & Switching
- Manejo de CLI
- Conceptos multidisciplinarios (Enterprise, Core, SP, Security, etc)



Cloud y servicios

- DevOps
- Arquitecturas distribuídas (contenedores, k8s)
- Proveedores de cloud
- Manejo de infraestructura





github.com/ponchotitlan
/witcom-2024-code-
is-the-new-cli



