DevNet Zone at Cisco Live!

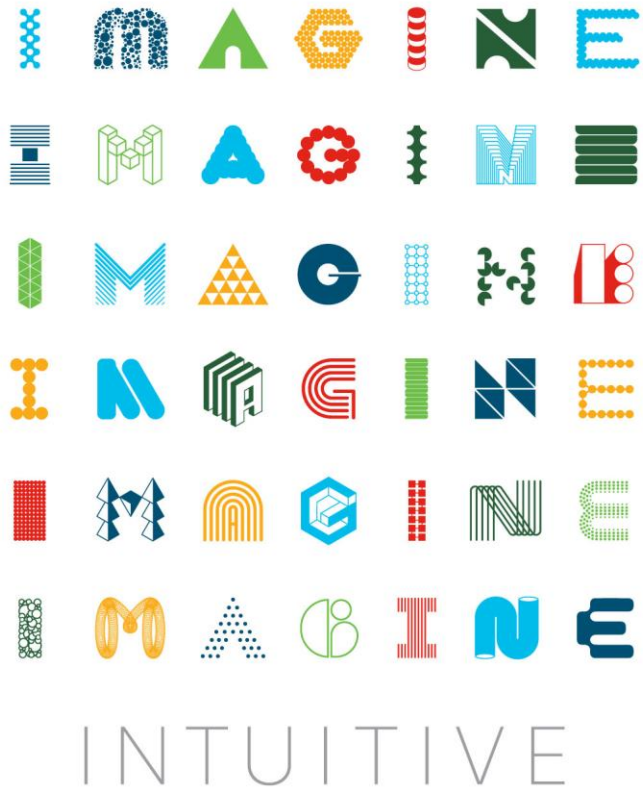
INTUITIVE

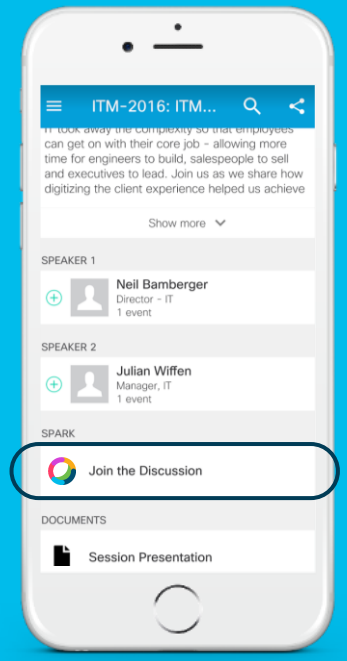
NetDevOps Engineer Everyday Skills

ConfigApprove Use Case

Gabriel Zapodeanu
Technology Solutions Architect, Cisco Systems,
@zapodeanu gzapodea@cisco.com github.com/gzapodea

DEVWKS -1301





cs.co/ciscolivebot#DEVWKS-1301

Cisco Webex Teams

Questions?

Use Cisco Webex Teams (formerly Cisco Spark) to chat with the speaker after the session

How

1. Open the Cisco Events Mobile App
2. Find your desired session in the “Session Scheduler”
3. Click “Join the Discussion”
4. Install Webex Teams or go directly to the team space
5. Enter messages/questions in the team space

https://t.me/learningnets

Gabriel Zapodeanu
Technology Solutions Architect,
Cisco Systems

Gabriel Zapodeanu is a Technology Solutions Architect, focused on network programmability. He is presently part of the Architecture and Engineering team, Cisco Global Partner Organization.

His work is focused on infrastructure programmability, partner enablement programs, APIs use case development across various technologies, as well as learning and certification programs.
After joining Cisco in 2006 he worked in the West Area Enterprise Organization, Oregon USA. His work includes large architectures for a variety of Enterprise customers from manufacturing, utilities, retail, transportation, healthcare, education, and financial industries.

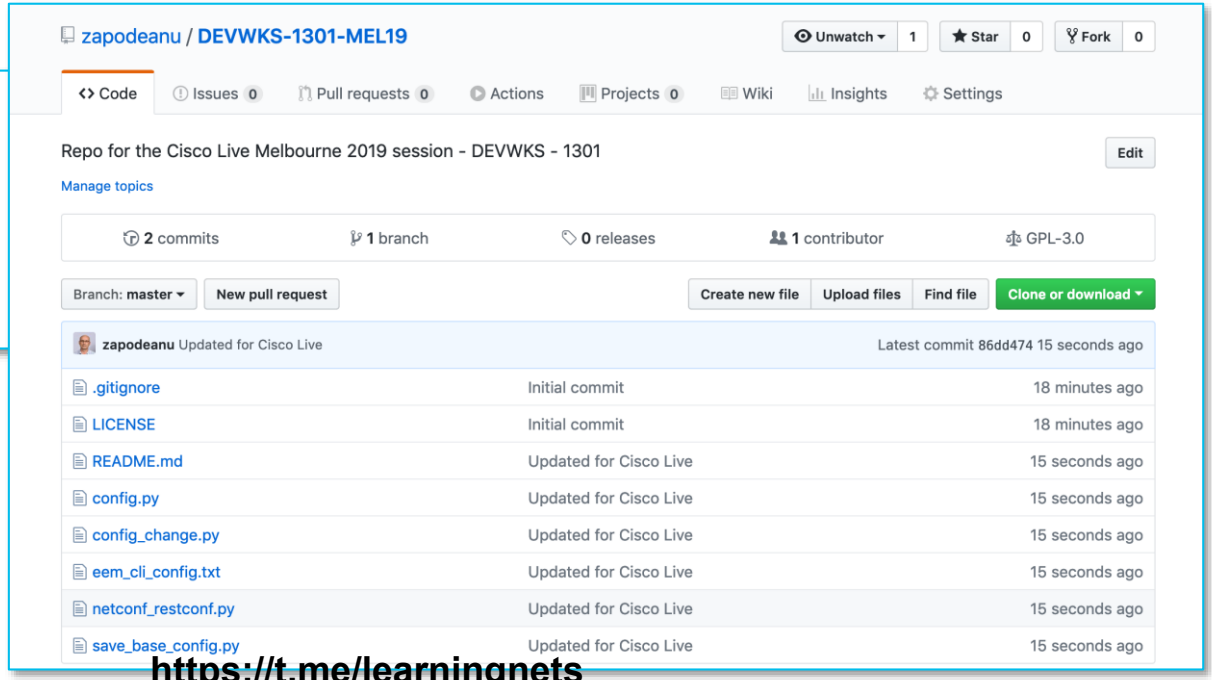
Gabriel completed his Cisco Business Architecture Certification #373. Cisco Live Distinguished Speaker 2014.
Recently, Gabriel co-authored the book "IOS XE Programmability: Automating Device Lifecycle Management"

Session Presentation and Code

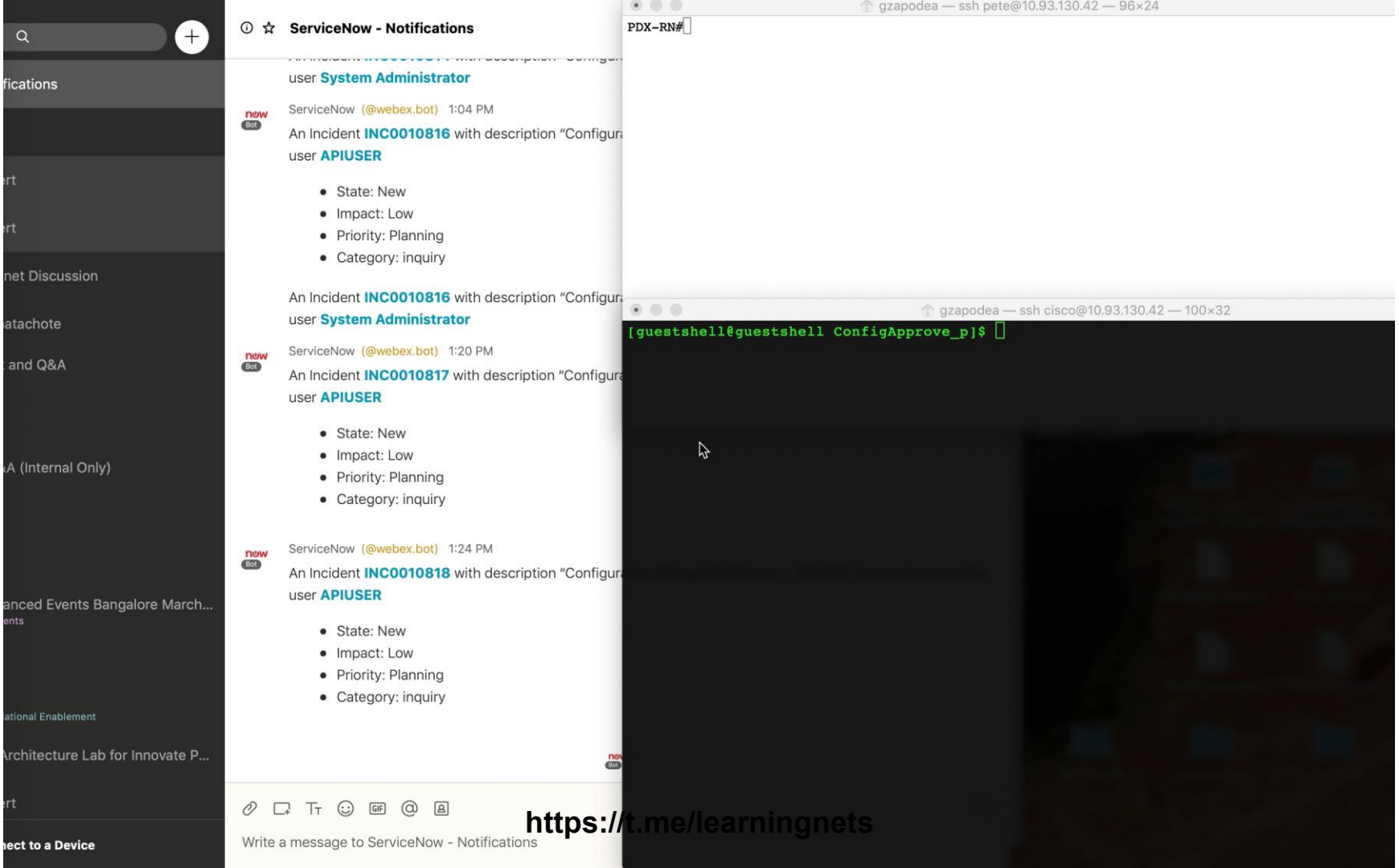
https://github.com/zapodeanu/DEVWKS-1301-MEL19

Desktop Folder
DEVWKS-1301

- Code
- Presentation



DEMO



Acronym Decoder / Technical Terms

- GS – Guest Shell
- IOX – Virtualization Manager
- DNS – Domain Name System
- sudo – Super User Do
- PIP – Tool for installing and managing Python packages (Pip Installs Packages)
- YUM – (Yellowdog Updater Modified) open source package management tool
- CentOS – Linux distribution derived from RHEL
- Bash – Bourne Again Shell, Linux standard shell
- PNP – Plug and Play
- CAF – Cisco Application Hosting Framework
- ZTP – Zero Touch Provisioning

Workshop Requirements

- DevNet account – you have this, use your CCO account
- Webex Teams account – let me know if you do not have this
- GitHub account – http://github.com
- ServiceNow developer account (optional)
- DevNet CSR1000V sandbox – provided or reserved by you: https://developer.cisco.com/site/sandbox/ –

  IOS XE Programmability NETCONF-RESTCONF-YANG
- VPN client – installed already on the provided laptops

Agenda

- NetDevOps
- IOS XE Guest Shell Configuration
- Device Management Using:
  - Python CLI
  - NETCONF
  - RESTCONF
- Run Apps on Guest Shell
- Embedded Event Manager
- ConfigApprove – ChatOps Use Case
- ConfigApprove Roadmap

Agenda

- NetDevOps
- IOS XE Guest Shell Configuration
- Device Management Using:
  - Python CLI
  - NETCONF
  - RESTCONF
- Run Apps on Guest Shell
- Embedded Event Manager
- ConfigApprove – ChatOps Use Case
- ConfigApprove Roadmap

What is NetDevOps?

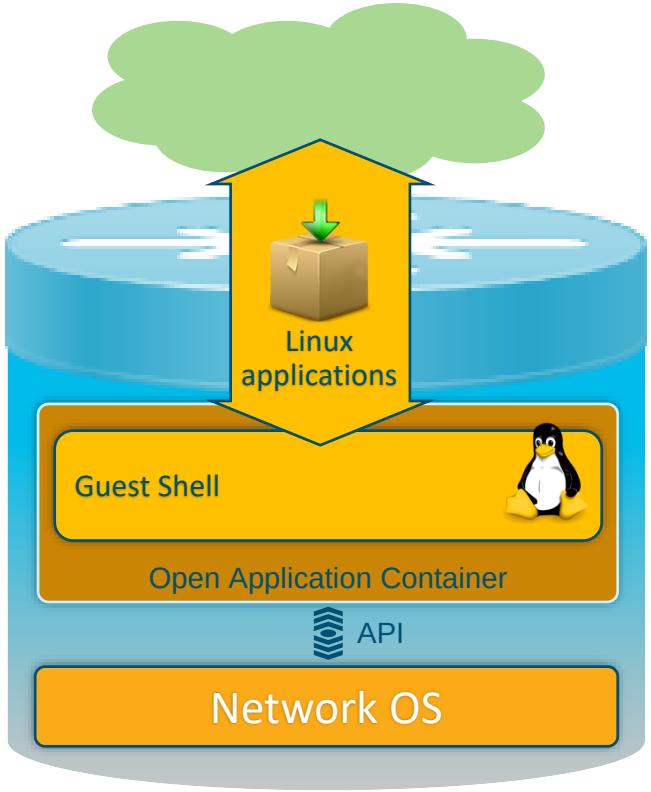
- Networking + Dev Ops
- DevOps values of automation and scalability applied to network infrastructure
- Starts with network automation
- Requires Networking and Programming skills

Agenda

- NetDevOps
- IOS XE Guest Shell Configuration
- Device Management Using:
  - Python CLI
  - NETCONF
  - RESTCONF
- Run Apps on Guest Shell
- Embedded Event Manager
- ConfigApprove – ChatOps Use Case
- ConfigApprove Roadmap

Guest Shell Application

- Linux Shell Environment on your device
- Integrated with IOS XE
- Maintains IOS XE system integrity
  - Isolated User Space
  - Fault Isolation
  - Resource Isolation
- On-box rapid prototyping
  - Device-level API Integration
  - Scripting (Python)
  - Linux Commands
- Application Hosting
- Integrates into your Linux workflow

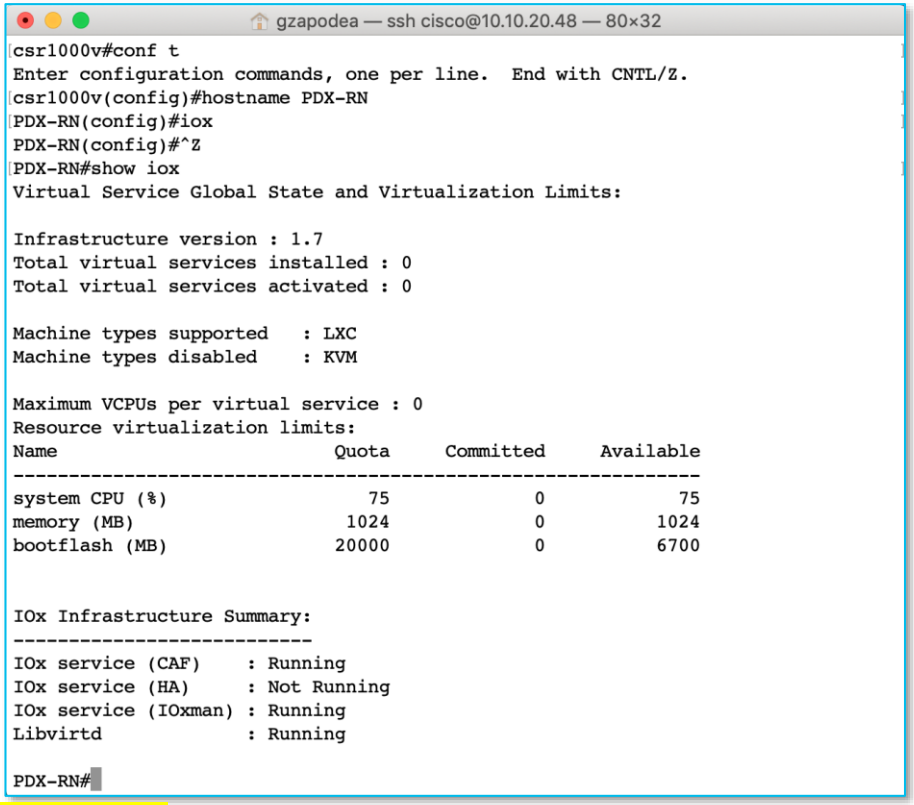


IOS XE Guest Shell Overview

- Guest Shell is a decoupled and secure environment running within a Linux Container (LXC)
- From within the Guest Shell the network admin has the following capabilities:
  - Access to the network over Linux network interfaces
  - Access to /bootflash
  - Device management using CLI, NETCONF and RESTCONF
  - The ability to install and run Linux applications
  - The ability to install and run on-box Python Scripts

Configure IOX

- IOX is the manager that handles guest shell and others apps
- Use the provided instructions to connect to your sandbox
- **ssh** to your CSR1000V router (10.10.20.48)
- Change the name of your device
- Enter the command iox in config mode
- It will take a couple of minutes for the IOX services to start running
- Verify IOX operations with the command **show iox**

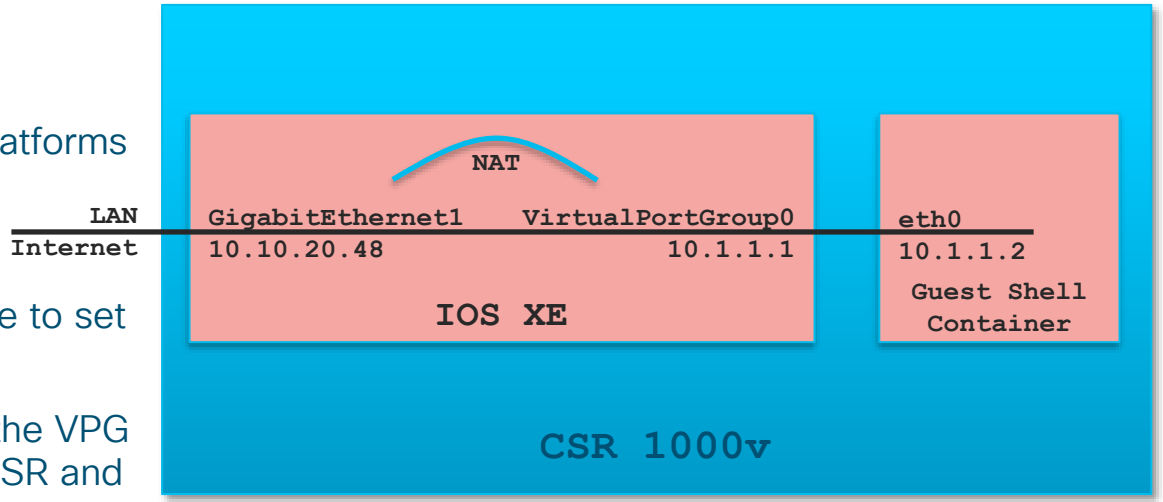


Note: Execute only highlighted commands

https://t.me/learningnets

Guest Shell Networking

- Guest Shell network access using:
  - management interface/ management vrf – Switches, Routers
  - VirtualPortGroup (VPG) for platforms without a management port
- We are going to use the VPG mode to set up Guest Shell networking
- The guest shell container will use the VPG to communicate with the IOS XE CSR and the Internet



Guest Shell Networking – continued

- Interface GigabitEthernet 1 is connected to the Internet for external access
- Need a VirtualPortGroup interface with static ip address
- NAT set up between VPG and the GigabitEthernet 1

Configure your CSR1000V with these commands:

```
aaa new-model
aaa authentication login default local
aaa authorization exec default local

ip name-server 208.67.222.222

interface VirtualPortGroup0
 ip address 10.1.1.1 255.255.255.0
 description used_for_GS_access
 ip nat inside

interface GigabitEthernet1
 ip nat outside

ip access-list standard GS_NAT_ACL
 permit 10.1.1.0 0.0.0.255

ip nat inside source list GS_NAT_ACL interface GigabitEthernet1 overload
```

Guest Shell Configuration

Enable Guest Shell and verify connectivity from IOS XE:

```
PDX-RN(config)#app-hosting appid guestshell
PDX-RN(config-app-hosting)#vnic gateway1 virtualportgroup 0 guest-interface 0 guest-ipaddress 10.1.1.2 netmask 255.255.255.0
gateway 10.1.1.1 name-server 208.67.222.222
PDX-RN#guestshell enable
Interface will be selected if configured in app-hosting
Please wait for completion
Guestshell enabled successfully
PDX-RN#show app-hosting list
App id                           State
---------------------------------------------------------
guestshell                       RUNNING
PDX-RN#guestshell run sudo ping 10.1.1.1
PING 10.1.1.1 (10.1.1.1) 56(84) bytes of data.
64 bytes from 10.1.1.1: icmp_seq=1 ttl=255 time=0.527 ms
^C
PDX-RN#guestshell ?
  destroy  Disable and uninstall the guest shell service package
  disable  Disable the guest shell service package
  enable   Enable the guest shell service
  run      Execute/run program in the guest shell
PDX-RN#show app-hosting detail
```

Note: Execute only highlighted commands

Guest Shell Configuration – continued

- Verify connectivity from Guest Shell to IOS XE and Internet

```
PDX-RN#guestshell run bash
[guestshell@guestshell ~]$ cat /etc/resolv.conf
nameserver 208.67.222.222
[guestshell@guestshell ~]$ ping 10.1.1.1
PING 10.1.1.1 (10.1.1.1) 56(84) bytes of data.64 bytes from 10.1.1.1:
icmp_seq=1 ttl=255 time=0.568 ms64 bytes from 10.1.1.1:
icmp_seq=2 ttl=255 time=0.407 ms64 bytes from 10.1.1.1:
icmp_seq=3 ttl=255 time=0.467 ms64 bytes from 10.1.1.1:
^C
[guestshell@guestshell ~]$ sudo ping 10.10.20.254
64 bytes from 10.10.20.254: icmp_seq=1 ttl=255 time=0.790 ms
64 bytes from 10.10.20.254: icmp_seq=2 ttl=255 time=0.880 ms
64 bytes from 10.10.20.254: icmp_seq=3 ttl=255 time=0.740 ms
^C
[guestshell@guestshell ~]$
```

Guest Shell Troubleshooting

- If not successful verify:
  - VirtualPortGroup and NAT Configuration
  - Guest Shell Configuration
  - Add a DNS configuration to the /etc/resolv.conf

```
PDX-RN#show run interface VirtualPortGroup 0
interface VirtualPortGroup0
 ip address 10.1.1.1 255.255.255.0
 ip nat inside

PDX-RN#show ip nat statistics
 Total active translations: 0 (0 static, 0 dynamic; 0 extended)
Outside interfaces:
  GigabitEthernet1
Inside interfaces:
   VirtualPortGroup0
Hits: 14768  Misses: 86
```

```
PDX-RN#guestshell run sudo ifconfig
  eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>.  mtu 1500
        inet 10.1.1.2  netmask 255.255.255.0  broadcast 10.1.1.255
        inet6 fe80::5054:ddff:fed0:1dbf  prefixlen 64
        ether 52:54:dd:d0:1d:bf  txqueuelen 1000  (Ethernet)
        RX packets 10012  bytes 12216093 (11.6 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5593  bytes 383829 (374.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
PDX-RN#guestshell run cat /etc/resolv.conf
nameserver 208.67.222.222
```

Install Python Packages

```
PDX-RN#guestshell run bash
[guestshell@guestshell ~]$ sudo pip install --upgrade pip
[guestshell@guestshell ~]$ sudo pip install requests
[guestshell@guestshell ~]$ sudo pip install requests[security]
[guestshell@guestshell ~]$ sudo pip install ncclient
[guestshell@guestshell ~]$ pip list
DEPRECATION: The default format will switch to columns in the future. You can use --
format=(legacy|columns) (or define a format=(legacy|columns) in your pip.conf under
the [list] section) to disable this warning.
…
cryptography (2.2.2)
pip (9.0.3)
requests (2.18.4)
…
[guestshell@guestshell ~]$ exit
```

Note: Execute only highlighted commands

Install Git and Nano on Guest Shell

- Install Git on CentOs (GS): **sudo yum install git -y**
- If the command completes without errors, you had git downloaded and installed.
- To double-check that it is working correctly, try running Git's built-in version check
- **git version** - git version 1.8.3.1
- Set-up Git with your info:
  - **git config --global user.name "Your GitHub Username "**
  - **git config --global user.email "you@example.com"**
  - **git config --global credential.helper "cache --timeout 3600"**
- Verify Git Config - **git config --list**
- Additional packages:
  - **sudo yum install curl-devel**
  - **sudo yum update -y nss curl libcurl**
- (May be required for git clone/pull operations on some platforms/software versions)
- Reference: https://www.digitalocean.com/community/tutorials/how-to-install-git-on-centos-7
- Install Nano: **sudo yum install nano -y**

Clone the Code Repo

```
PDX-RN#guestshell run bash
[guestshell@guestshell ~]$ cd /bootflash

[guestshell@guestshell bootflash]$ git clone 'https://github.com/zapodeanu/DEVWKS-1301-MEL19.git'
Cloning into 'DEVWKS-1301-MEL19'...
Username for 'https://github.com': gzapodea
Password for 'https://gzapodea@github.com':
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 13 (delta 1), reused 8 (delta 0), pack-reused 0
Unpacking objects: 100% (13/13), done.

[guestshell@guestshell bootflash]$ ls /bootflash/DEVWKS-1301-MEL19
LICENSE    config.py         eem_cli_config.txt   save_base_config.py
README.md  config_change.py  netconf_restconf.py
[guestshell@guestshell bootflash]$ exit
```


Guest Shell – Other Resources

- File operations IOS XE:
  - mkdir flash:/path/to/folder
  - delete /force /recursive flash:/path/to/folder
  - dir flash:
  - more flash:/path/to/file (display content of file)
- File operations Guest Shell (sometimes sudo is required)
  - mkdir /bootflash/path/to/folder
  - rm -rf /bootflash/path/to/folder
  - ls /bootflash/
  - cat /bootflash/path/to/file (display the content of file)
- Verify the IOS XE Programmability Guide for additional configurations (for example proxy configurations) and troubleshooting options

Agenda

- NetDevOps
- IOS XE Guest Shell Configuration
- Device Management Using:
  - Python CLI
  - NETCONF
  - RESTCONF
- Run Apps on Guest Shell
- Embedded Event Manager
- ConfigApprove – ChatOps Use Case
- ConfigApprove Roadmap

Python CLI Module

- Allows the user's Python scripts to run IOS CLI commands on the host device
- Interactive Python Prompt
  - guestshell run python
- Python Script
  - guestshell run python /bootflash/path/to/file.py

Device Management from Guest Shell Using the Interactive Python Prompt

```
PDX-RN#guestshell run python
Python 2.7.5 (default, Jun 17 2014, 18:11:42)
[GCC 4.8.2 20140120 (Red Hat 4.8.2-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import cli
>>> from cli import execute
>>> output = execute("show run | in hostname")
>>> print output
hostname PDX-RN
>>> print(execute("show interface Gi1"))
GigabitEthernet1 is up, line protocol is up
  Hardware is CSR vNIC, address is 0050.56ac.232f (bia 0050.56ac.232f)
  Description: DON'T TOUCH ME
  Internet address is 10.10.20.48/24
  MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
      reliability 255/255, txload 1/255, rxload 1/255  Encapsulation ARPA, loopback not set
…
>>> exit()
```

Note: Execute only highlighted commands

Device Management from Guest Shell using RESTCONF

```
PDX-RN#guestshell run bash
[guestshell@guestshell ~]$ cd /bootflash/DEVWKS-1301-MEL19/
[guestshell@guestshell DEVWKS-1301-MEL19]$ python netconf_restconf.py
Interface Operational Status via NETCONF:
up
Interface Operational Data via RESTCONF:
{
    "speed" : "1024000000" ,
    "statistics" : {
        "out-octets" : "4623697" ,
        "out-unicast-pkts" : "56354" ,
        "in-unicast-pkts" : "108530" ,
        …..} ,
    "name" : "GigabitEthernet1" ,
    "oper-status" : "up" ,
"admin-status" : "up" ,
    "if-index" : 1
}
Device Hostname via RESTCONF:
PDX-RN
[guestshell@guestshell DEVWKS-1301-MEL19]$ exit
```

Read and execute the file **netconf_restconf.py**

Agenda

- NetDevOps
- IOS XE Guest Shell Configuration
- Device Management Using:
  - Python CLI
  - NETCONF
  - RESTCONF
- Run Apps on Guest Shell
- Embedded Event Manager
- ConfigApprove – ChatOps Use Case
- ConfigApprove Roadmap

Prepare to Execute Python Apps

```
PDX-RN#wri mem
PDX-RN#guestshell run bash
[guestshell@guestshell ~]$ cd /bootflash/
[guestshell@guestshell bootflash]$ mkdir CONFIG_FILES
[guestshell@guestshell bootflash]$ ls
CONFIG_FILES
DEVWKS-1301-MEL19
…
[guestshell@guestshell bootflash]$ cd DEVWKS-1301-MEL19
Get from GitHub the latest application code:
[guestshell@guestshell DEVWKS-1301-MEL19]$ git pull
Read and execute the initialize script:
[guestshell@guestshell DEVWKS-1301-MEL19]$ python save_base_config.py
```

Download the latest code

Save baseline configuration
save_base_config.py

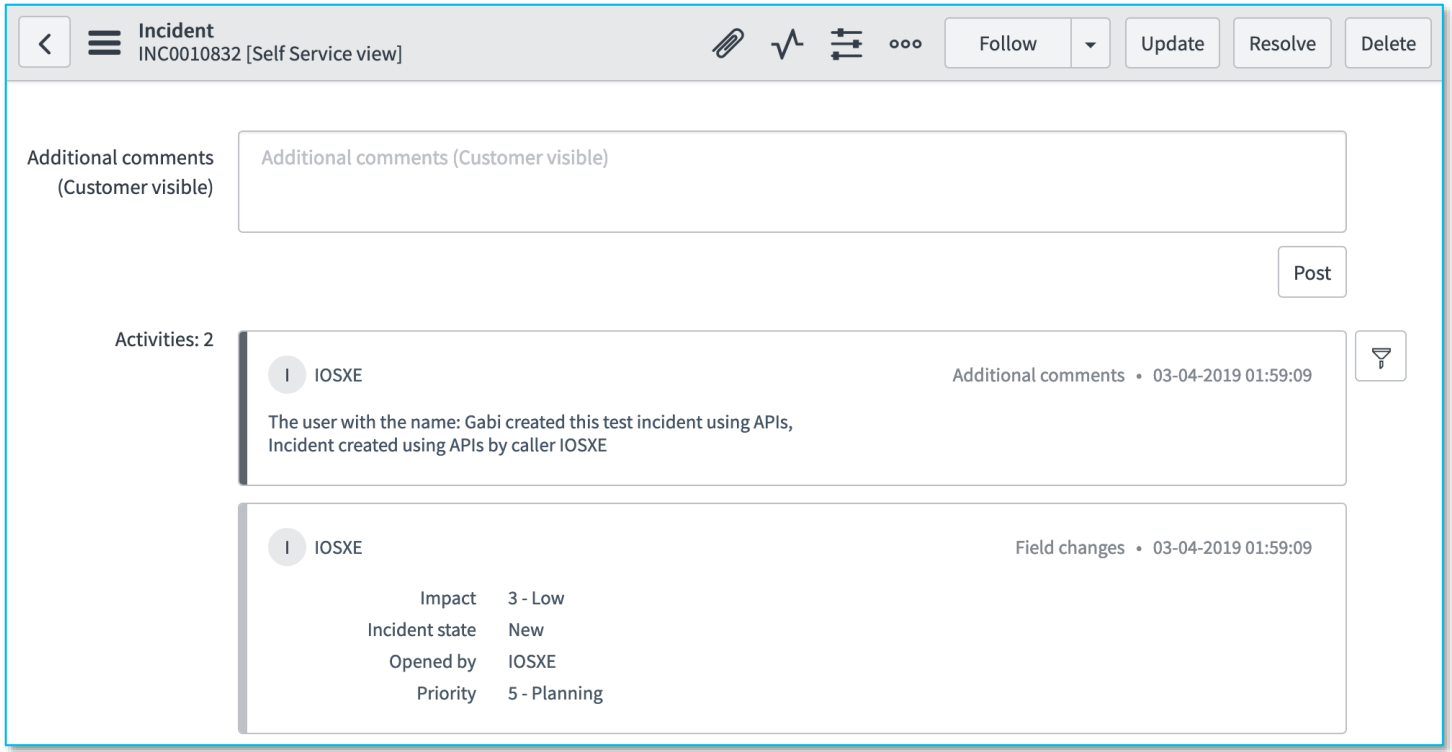
Python Scripts on Guest Shell

Create a new test ServiceNow incident

```
PDX-RN#guestshell run bash
[guestshell@guestshell ~]$ cd /bootflash/DEVWKS-1301-MEL19/
[guestshell@guestshell DEVWKS-1301-MEL19]$ python create_incident.py
Your name is :
Gabi
APIUSER ServiceNow sysid is: e1b1231e4f4213004419ff6f9310c765

ServiceNow REST API call response: 201

End Application Run
[guestshell@guestshell DEVWKS-1301-MEL19]$ more create_incident.py
# developed by Gabi Zapodeanu, TSA, Global Partner Organization
import requests
import json
import logging
….
```

Execute and Read the file **create_incident.py**

ServiceNow Incident



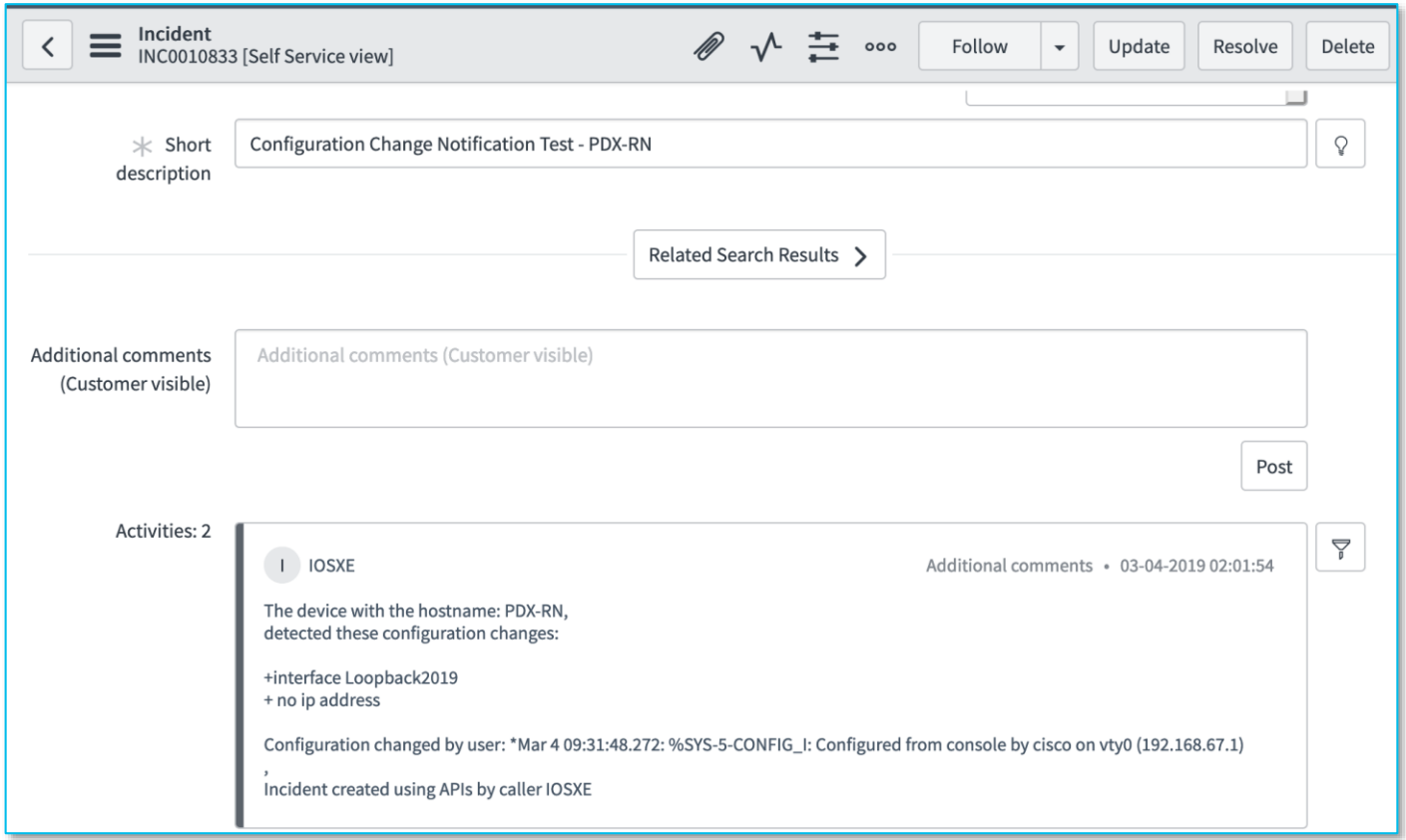
Python Scripts on Guest Shell (continued)

Make a configuration change and create a new ServiceNow incident

```
PDX-RN#configure t
Configure something
PDX-RN#guestshell run bash
[guestshell@guestshell ~]$ cd /bootflash/DEVWKS-1301-MEL19/
[guestshell@guestshell DEVWKS-1301-MEL19]$ python config_change_incident.py
-interface Loopback2019
- description CL MEL
- no ip address
('Hostname: ', 'PDX-RN')
('Created new ServiceNow incident with the number: ', 'INC0010822')
End Application Run
[guestshell@guestshell DEVWKS-1301-MEL19]$ more config_change_incident.py
# developed by Gabi Zapodeanu, TSA, Global Partner Organization
import requests
import json
import logging
….
```

Execute and Read the file **config_change_incident.py**

ServiceNow Incident with Configuration Change



Agenda

- NetDevOps
- IOS XE Guest Shell Configuration
- Device Management Using:
  - Python CLI
  - NETCONF
  - RESTCONF
- Run Apps on Guest Shell
- Embedded Event Manager
- ConfigApprove – ChatOps Use Case
- ConfigApprove Roadmap

EEM Python Policies

- Offers the ability to monitor events and take action when the monitored events occur or when a threshold is reached
- Embedded Event Manager (EEM) uses software programs known as **event detectors** to determine when an EEM event occurs
- Embedded Event Manager (EEM) policies support Python scripts
- Configured EEM policies run within the Guest Shell

EEM Configuration

Configure EEM on your device the commands from the **eem_cli_config.txt file**

```
PDX-RN#copy run start
Destination filename [startup-config]?
Building configuration...
[OK]
PDX-RN# delete flash:/CONFIG_FILES/base-config
PDX-RN#configure terminal
event manager applet config_change
 event syslog pattern "SYS-5-CONFIG_I"
 action 1.0 cli command "enable"
 action 2.0 cli command "guestshell run python /bootflash/DEVWKS-1301-MEL19/config_change_approve.py"
 action 3.0 cli command "exit"
 action 4.0 cli command "end"
end
PDX-RN#show run | b event
event manager applet config_change
 event syslog pattern "SYS-5-CONFIG_I"
 action 1.0 cli command "enable"
 action 2.0 cli command "guestshell run python /bootflash/DEVWKS-1301-MEL19/config_change_approve.py"
 …
PDX-RN#copy run start
```

event detector

Agenda

- NetDevOps
- IOS XE Guest Shell Configuration
- Device Management Using:
  - Python CLI
  - NETCONF
  - RESTCONF
- Run Apps on Guest Shell
- Embedded Event Manager
- ConfigApprove – ChatOps Use Case
- ConfigApprove Roadmap

ConfigApprove Workflow

- User makes IOS XE device configuration change
- Syslog triggers EEM Python script execution in Guest Shell
- The config_change.py script will:
  - Detect if the configuration changed and what changed
  - Collect the device hostname using RESTCONF
  - Identify the user that made the change using Python CLI
  - Create WEBEX_TEAMS room using REST APIs, invite Approver to room, post the above information to ask for approval
  - If changes approved, save new configuration as baseline
  - If not approved or no response, rollback to the previous baseline configuration
  - Close the WEBEX_TEAMS room in 30 seconds
  - Create ServiceNow incident to record all of the above information

ConfigApprove APIs

- The Python application running on Guest Shell will use:
  - RESTCONF to connect to the host device
  - Python CLI to collect additional information from the host device
  - REST APIs to ServiceNow to create and update incidents
  - REST APIs to Webex Teams to create and delete rooms, post and read messages
  - File operations
  - Logging to file

Prepare to Execute 1st Time the App

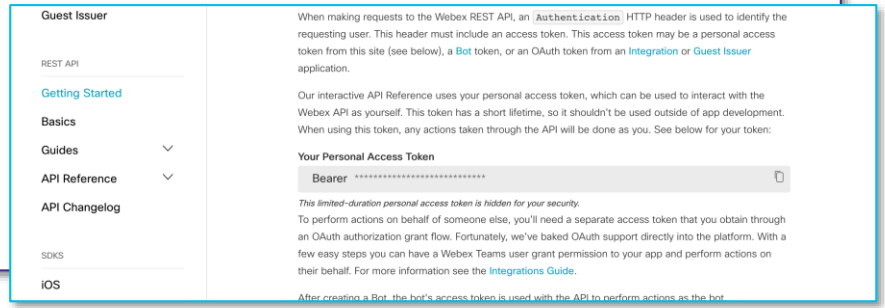
```
PDX-RN#wri mem
PDX-RN#guestshell run bash
[guestshell@guestshell ~]$ cd /bootflash/
[guestshell@guestshell bootflash]$ ls
CONFIG_FILES
DEVWKS-1301-MEL19
…
[guestshell@guestshell bootflash]$ cd DEVWKS-1301-MEL19
Get from GitHub the latest application code:
[guestshell@guestshell DEVWKS-1301-MEL19]$ git pull
Read and execute the initialize script:
[guestshell@guestshell DEVWKS-1301-MEL19]$ python save_base_config.py
```

Edit Configuration File

- edit the config.py file located /bootflash/DevNet... :
- nano or sudo vi config.py

```
# developed by Gabi Zapodeanu, TSA, GPO, Cisco Systems

# This file contains the WEBEX_TEAMS Auth, ServiceNow Dev account info, router info

WEBEX_TEAMS_URL = 'https://api.ciscospark.com/v1'
WEBEX_TEAMS_AUTH = 'Bearer ' + 'your WEBEX_TEAMS Token'
WEBEX_TEAMS_SPACE = 'Config Change Alert'
WEBEX_TEAMS_MEMBER = 'gzapodea@cisco.com'

SNOW_URL = 'https://dev48476.service-now.com/api/now'
SNOW_USER = 'APIUSER'
SNOW_PASS = 'Clive19!'
SNOW_INSTANCE = 'dev48476'

HOST = '10.10.20.48'
USER = 'cisco'
PASS = 'cisco_1234!'
```

Find your Webex Teams Token from
http://developer.webex.com
Login > Documentation > Getting Started



https://t.me/learningnets

Verification

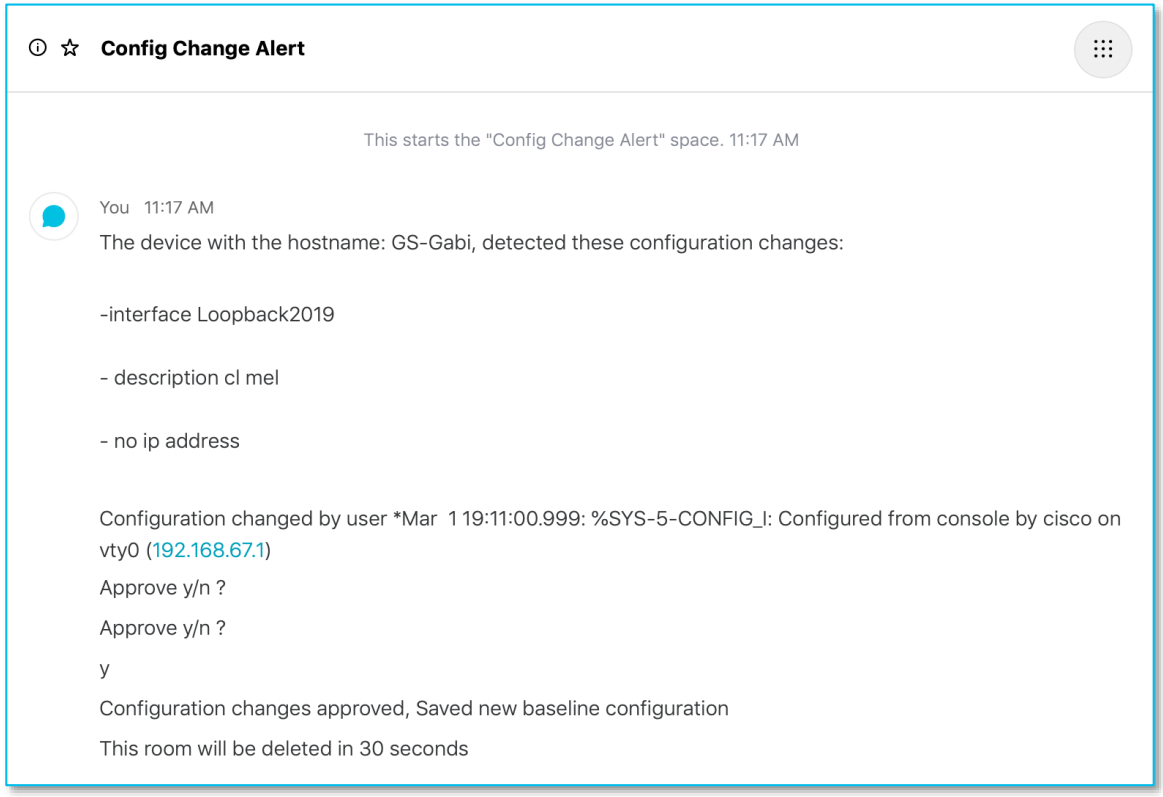
- Guest Shell commands:

```
[guestshell@guestshell]$ cd /bootflash
[guestshell@guestshell bootflash]$ ls
CONFIG_FILES
DEVWKS-1301-MEL19
[guestshell@guestshell bootflash]$ ls CONFIG_FILES/
base-config
[guestshell@guestshell bootflash]$ cat DEVWKS-1301-MEL19/config.py
# developed by Gabi Zapodeanu, TSA, GPO, Cisco Systems
# This file contains the WEBEX_TEAMS Auth, ServiceNow Dev account info, router info
WEBEX_TEAMS_URL = 'https://api.ciscospark.com/v1'
WEBEX_TEAMS_AUTH = 'Bearer ' + 'NzRiZGU0M2UtOWJjNy00MzQwLWE4MjUtO…'
WEBEX_TEAMS_SPACE = 'Config Change Alert'
WEBEX_TEAMS_MEMBER = 'gzapodea@cisco.com'
…
exit
```

Let's Have Some Fun!

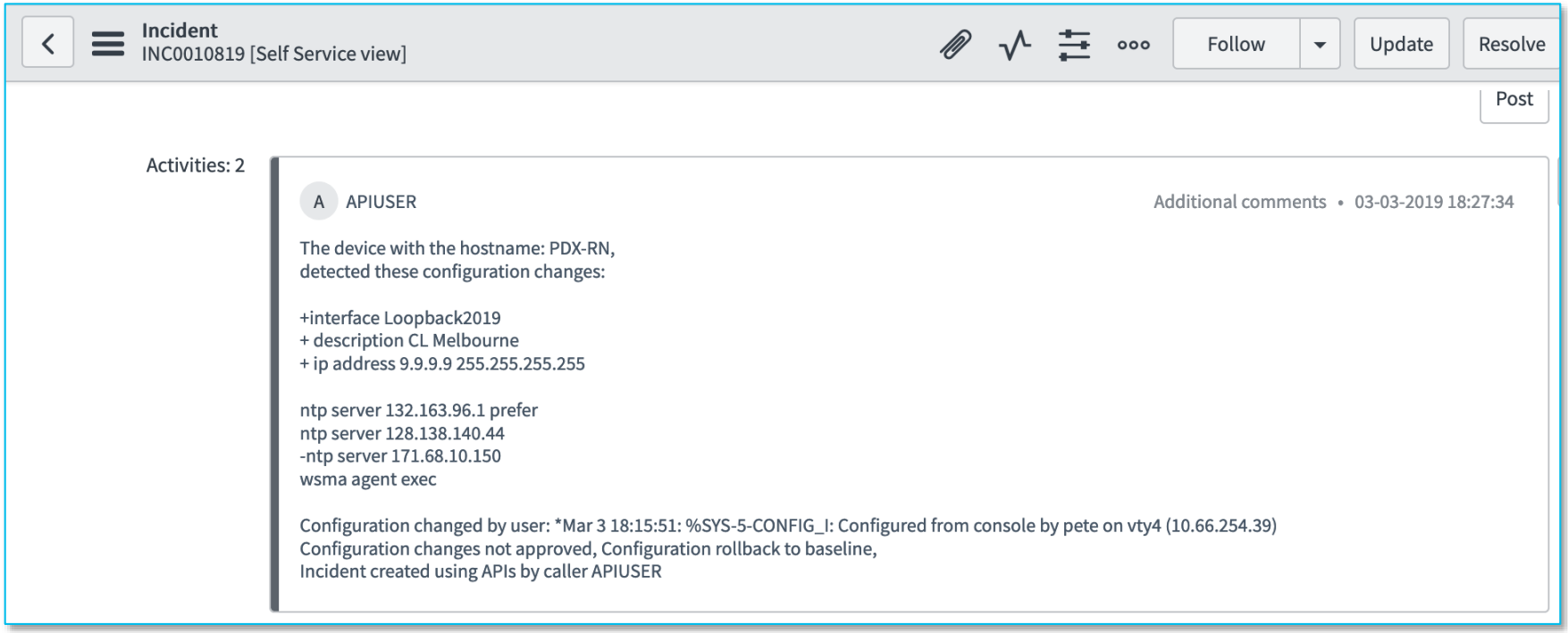
```
Make some configurations changes
PDX-RN#conf t
PDX-RN(config)#interface loopback 2019
PDX-RN(config-if)#description CL Melbourne
PDX-RN(config-if)#ip add 18.18.18.18 255.255.255.255
PDX-RN(config-if)#end
PDX-RN#
*Nov 27 21:56:03.694: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Loopback2018, changed state to up
*Nov 27 21:56:03.694: %LINK-3-UPDOWN: Interface Loopback2018, changed state to up
*Nov 27 21:56:20.659: %DMI-5-SYNC_NEEDED: R0/0: syncfd: Configuration change
requiring running configuration sync detected - 'ip address 18.18.18.18
255.255.255.255'. The running configuration will be synchronized  to the NETCONF
running data store.
*Nov 27 21:56:22.884: %SYS-5-CONFIG_I: Configured from console by cisco on vty1
(192.168.16.1)
```


Webex Teams Space



https://t.me/learningnets

ServiceNow



Agenda

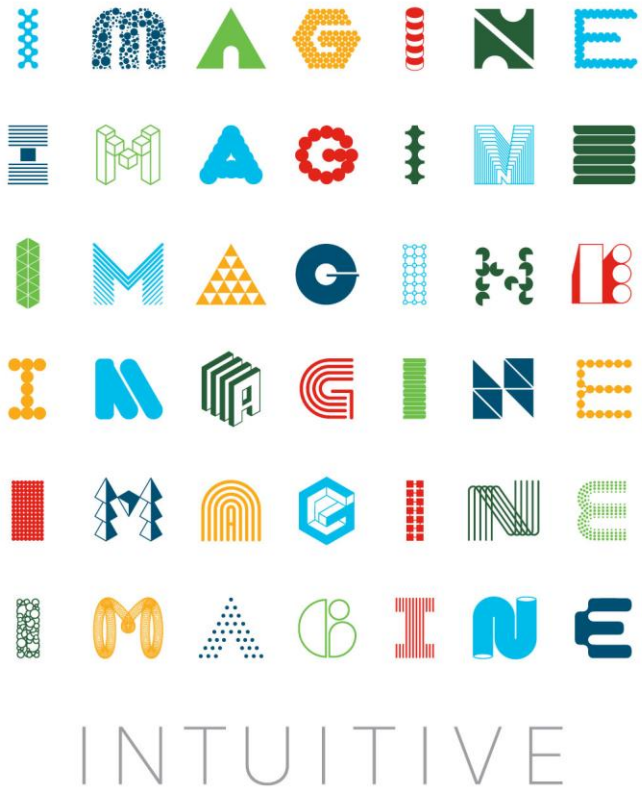
- NetDevOps
- IOS XE Guest Shell Configuration
- Device Management Using:
  - Python CLI
  - NETCONF
  - RESTCONF
- Run Apps on Guest Shell
- Embedded Event Manager
- ConfigApprove - ChatOps Use Case
- ConfigApprove Roadmap

https://t.me/learningnets

ConfigApprove Roadmap

- Integration with Cisco DNA Center
- Identify physical location of device
- Trigger DNA C device sync if configuration approved
- Business rule integrations:
  - ”No configuration changes during business hours”
- Rollback changes if configuration does not pass validations:
  - ”No duplicate IP addresses”
- Automatic code sync with github
- Voice/text notifications

Q & A

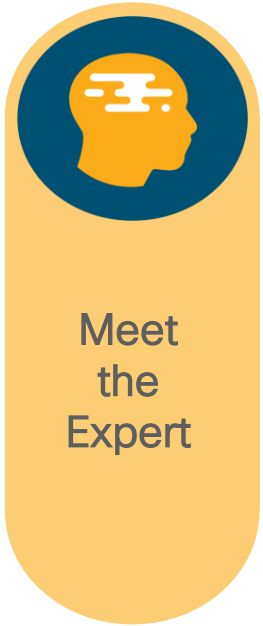


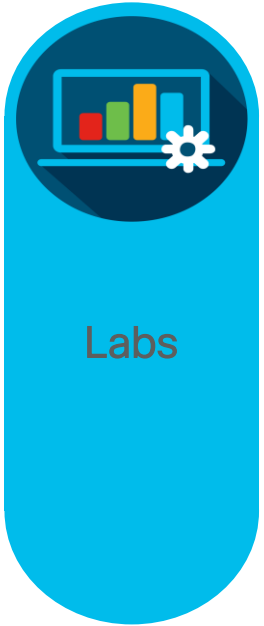
DevNet Code Exchange

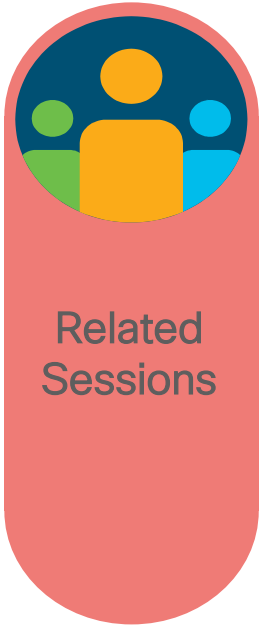
Get your code in front of the DevNet Community

developer.cisco.com/codeexchange

Continue your education







https://t.me/learningnets

Complete Your Online Session Evaluation

- Give us your feedback and receive a complimentary **Cisco Live 2019 Power Bank** after completing the overall event evaluation and 5 session evaluations.
- All evaluations can be completed via the Cisco Live Melbourne Mobile App.
- Don't forget: Cisco Live sessions will be available for viewing on demand after the event via https://ciscolive.cisco.com/

Join the Cisco Developer Community

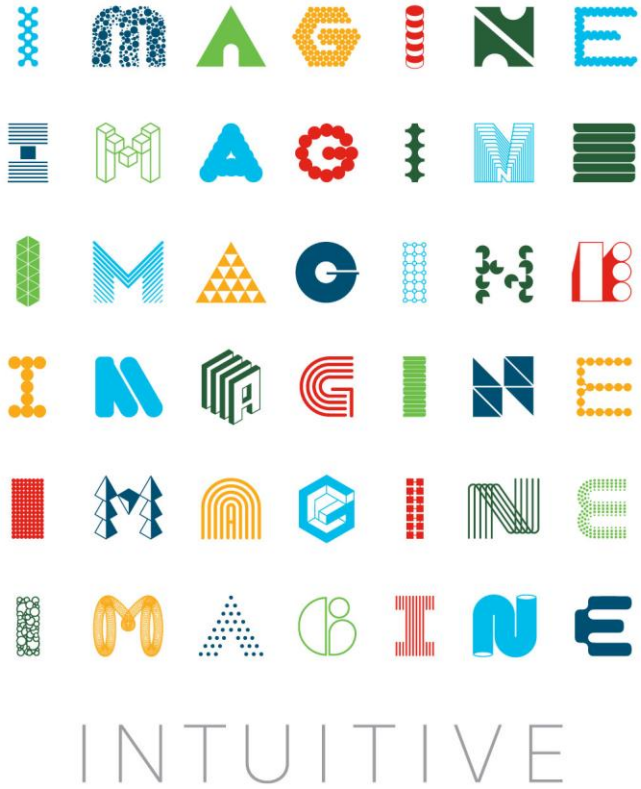
There is always something new to learn about programmable infrastructure and developer tools, using Cisco APIs.

Join the Cisco Developer Community for free.
https://developer.cisco.com/join/clmel19

DevNet Zone at Cisco Live!

https://t.me/learningnets

Thank you



DevNet Zone at Cisco Live!

INTUITIVE