Build your own Wi-Fi automation lab

Day 1 presentations

Andreas Koksrud / Telenor Bedrift

References / inspiration

https://github.com/CiscoDevNet/yangsuite/
https://postman.com
https://docs.ansible.com/ansible/latest/collections/cisco/ios/ios_facts_module.html
https://www.wifireference.com/2020/01/14/viewing-network-telemetry-from-the-catalyst-9800-with-grafana/
https://grafana.com/grafana/dashboards/13462-device-health-monitoring/
https://grafana.com/grafana/dashboards/12468-catalyst-9800-client-stats/
https://wirelessisfun.wordpress.com/2020/12/10/network-telemetry-data-and-grafana-part-1-the-advanced-netconf-explorer/
https://python.org
https://codeium.com

Copyright

Prerequisites

- Cisco.com account with
  - Access to downloading WLC (9800-CL at https://software.cisco.com)
  - Access to DevNet Sandbox (https://devnetsandbox.cisco.com/)
- Postman account (https://postman.com)
- Windows laptop with administrative privileges

- Without these, parts of the Deep Dive might be difficult to complete, or steps might differ severely

Communications

- WebEx space: Wi-Fi automation lab
- Please help each other
- Sharing is caring ☺

Agenda

Pre-lab

- Install VirtualBox
- Install Cisco 9800-CL
- Install Ubuntu Server w/Docker
- Install Postman
- Install VS Code

Day 1

- Finish the pre-lab tasks
- Get to know the lab environment
- Configure your AP
- Connect VS Code to Ubuntu
- Install and explore Ansible
- Explore Python automation
- Install and explore YANG Suite
- Explore Postman
- Install and explore Grafana

Day 2

- In-depth explore a topic of choice
  - Grafana / TIG-stack
  - Grafana Cloud
  - Ansible
  - Python
  - Other vendors (MIST? Meraki?)

Agenda - About Day 2 choices...

- Choose a track
  - Horiontal - A taste of everything (try to cut at 1h for each)
  - Vertical - 3h deep dive into one of the topics

Ansible

- Run a CLI command
- CLI configuration
- Using Jinja2 templates
- Using RESTCONF
- Write your own module
- RESTCONF + own module
- Organizing projects
- … more to come

Python

- Using Netmiko
- Using RESTCONF
- Get metrics, draw a graph
- Working with AI companions
- Using .env files
- Nornir automation framework
- More Nornir
- … more to come

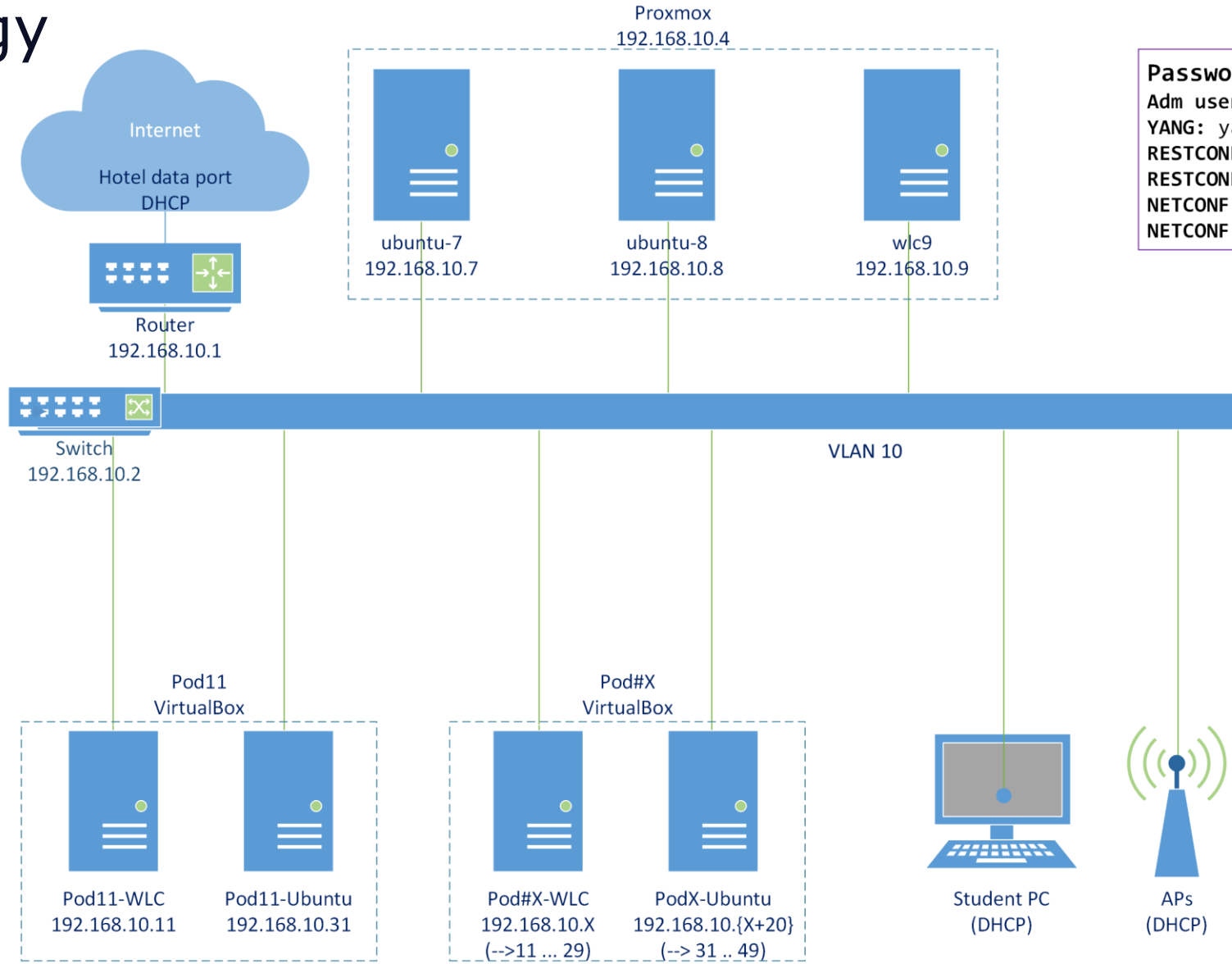
Grafana

- Working with syslog
- Finish building your own TIG stack from day 1
- Extended C9800 dashboard
- … building more
- … and improving even more

Scope

- In scope
  - Getting started with various systems/languages/solutions for lab purposes
  - Get a lab environment up and running on your own laptop
  - Some nice examples to try various aspects of automation
  - Inspiring you to explore deeper on your own
- Out of scope topics
  - Git
  - Learning the languages (Ansible, Python, InfluxQL, etc)
  - Deploying the systems for production use

Topology



Passwords to common devices

Adm user: devnet-adm / ChangeMe2024!
YANG: yang / yangyang
RESTCONF (RW): restconf-adm / restconf-pass
RESTCONF (RO): restconf-read / restconf-read
NETCONF (RW): netconf-adm / netconf-pass
NETCONF (RO): netconf-read / netconf-read

IP plan

Range	Usage
192.168.10.1-10	Static adm IPs
192.168.10.11-30	Pod#X WLC
192.168.10.31-50	Pod#X Ubuntu
192.168.10.51-70	Reserved static
192.168.10.71-250	DHCP range

POD addresses

Pod#	WLC IP	Ubuntu IP
11	192.168.10.11	192.168.10.31
12	192.168.10.12	192.168.10.32
13	192.168.10.13	192.168.10.33
14	192.168.10.14	192.168.10.34

(... WLC IP = Pod# + 20 for all...)

About us

- Andreas Koksrud
- Telenor
- Wi-Fi & Automation nerd
- M.Sc Communications technology
  + a bunch of Cisco certs)

- Kjetil Teigen Hansen
- Conscia
- Wi-Fi, monitoring and RF nerd
- CWNE#504
  + a bunch of Cisco certs

Automation in the Wi-Fi lifecycle

- Some examples of automation in different phases
- Prepare
  - Python scipt for information gathering / network mapping / device walking
- Plan
  - Python script for large-number analysis of devices (and/or maybe Excel is better for parts of it)
- Design
  - Prepare data to use in the Implement phase
- Implement
  - Python script for creating IP pools, or creating site hierarcy etc in DNAC
  - Ansible and/or some vendor specific Zero-Touch provisioning variants for device config based on some source of truth
- Operate
  - Write Python scripts for changes that involve large number of devices
  - Ansible for repeating tasks
  - Grafana for real-time troubleshooting
- Optimize
  - Grafana for monitoring/graphing/analysis
  - Python scripts or Ansible playbooks to analyze possible optimizations (APs on 100Mbit, hostnames of neighbors that differ, etc)

VirtualBox - Type 1 vs Type 2 hypervisors

Type 1

- Bare-metal (run directly on hardware)
- Superior performance
- For production use

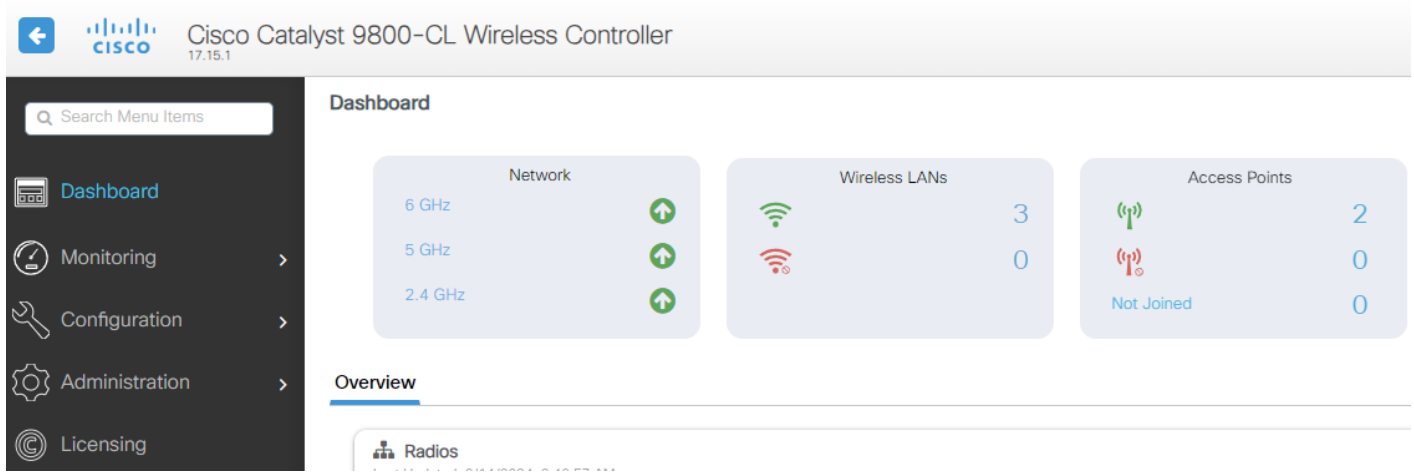
Examples
- VMWare ESXi
- Proxmox
- KVM
- Xen
- Nutanix

Type 2

- On top of another OS
- Flexible, can run on your laptop
- For lab purposes or at least non-24/7

Examples
- VirtualBox
- VMWare Workstation
- Hyper-V
- Parallels

WLAN Controllers



- This lab use Cisco 9800-CL WLC
- Project the knowledge to your platform of choice
  - Other vendors WLAN Controllers or similar concepts
  - Cloud-managed solutions often have good APIs
  - Ansible, Python, etc have lots of collections

Ubuntu Server

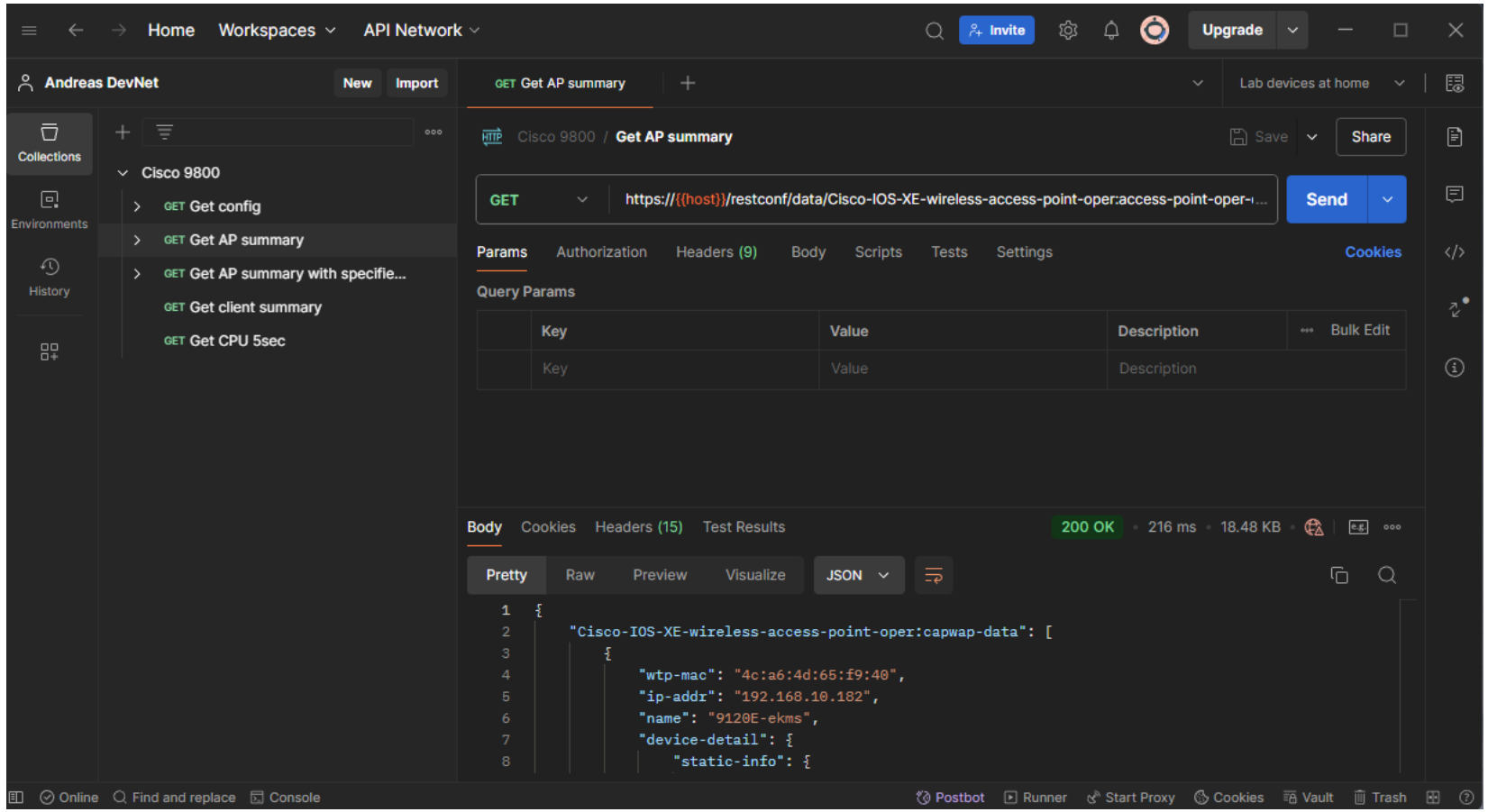
- This lab use Ubuntu Server
- Most stuff can be done directly on your own laptop
  - Currently Ansible can not run directly on Windows
  - TIG stack ... maybe(?)
- I like to play with stuff using a server, I find it easier to transfer to live systems
- WSL (Windows Subsystem for Linux) is a very good alternative for Python and Ansible. For Grafana it might be more tinkering

Docker

- This lab use Docker as the container platform
  - Pre-made packages for TIG stack and YANG Suite
  - Could just as well be using Kubernetes etc

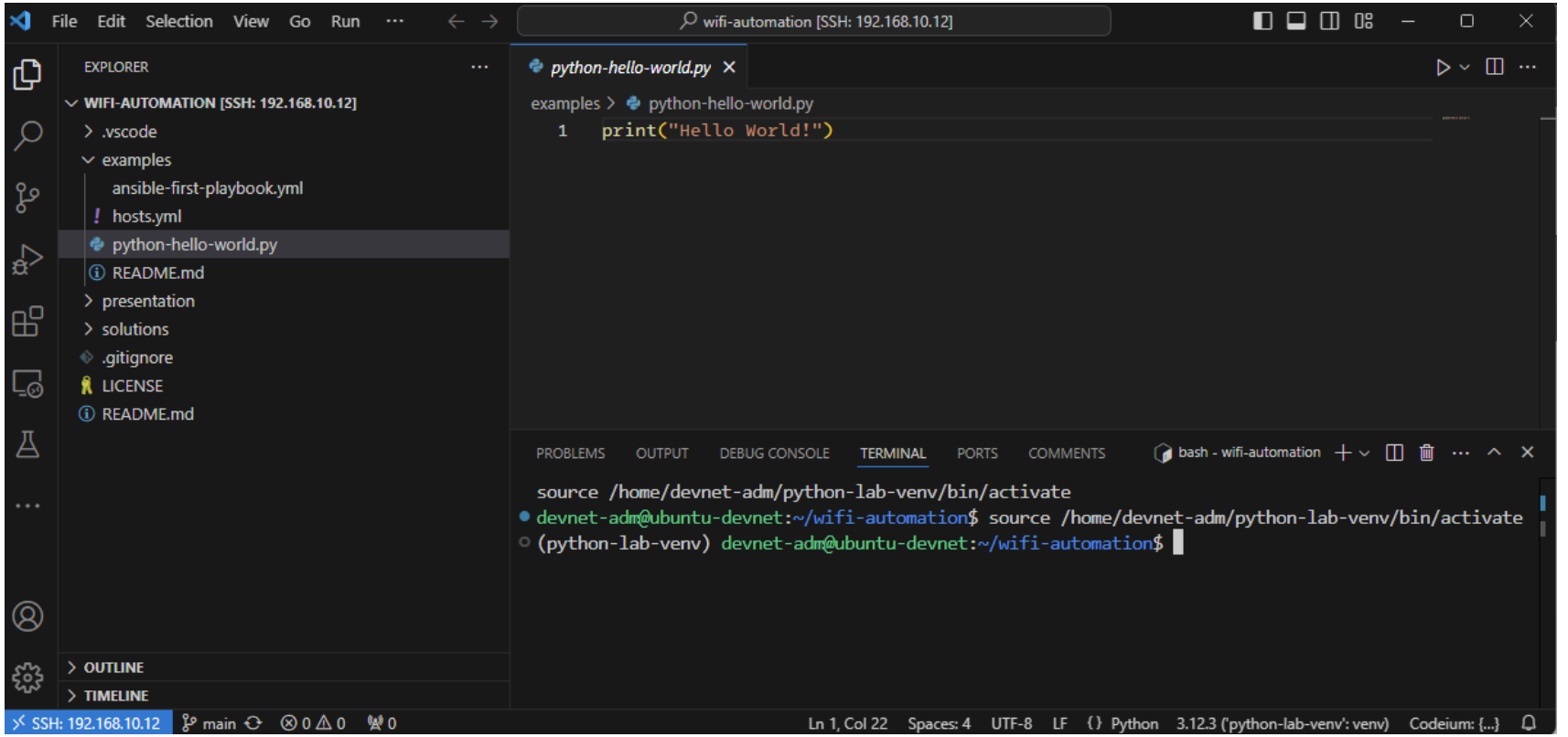
Postman

- Postman is a "go-to" tool for RESTCONF APIs. Some examples for automation are
  - Exploring RESTCONF calls
  - Checking the datastructure you get in return
  - Validating the calls to devices before implementing in Python, Ansible, etc
  - Get example code in your preferred language for that specific RESTCONF call



VS Code

- This lab use VS Code as text editor / development environment
  - Some other popular alternatives for all or parts of the process
    - VS Codium
    - Atom
    - Notepad++
    - Jupyter Notebook
    - Anaconda ecosystem



YANG Suite

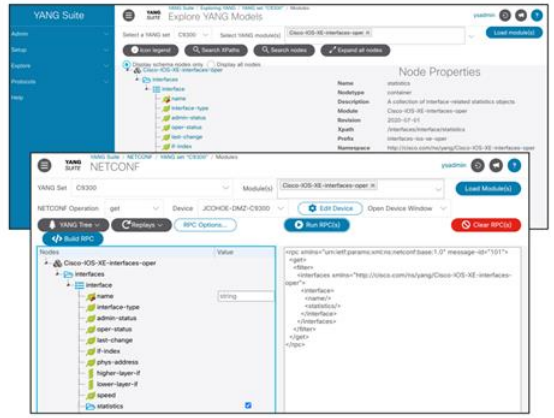
- Testing and validation environment for YANG related tasks
- Install as a docker container
- We will use this to explore the world of YANG models present in the Cisco 9800 WLC

Cisco YANG Suite

YANG API Testing and Validation Environment

Construct and test YANG based APIs over NETCONF, RESTCONF, gRPC and gNMI

IOS XE / IOS XR / NX OS platforms



Now Available !

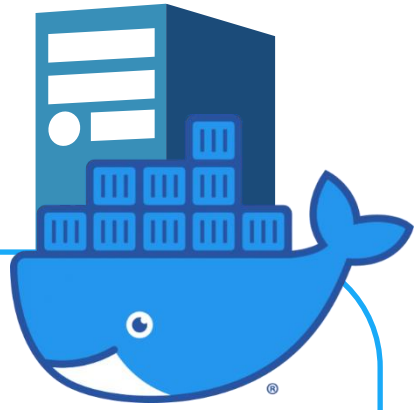
developer.cisco.com/yangsuite

github.com/CiscoDevNet/yangsuite

TIG Stack

- Bla bla about what is TIG stack

gRPC / TCP 57000



telegraf™

metrics goes to the database

influxdb TCP 8086

Grafana to visualize

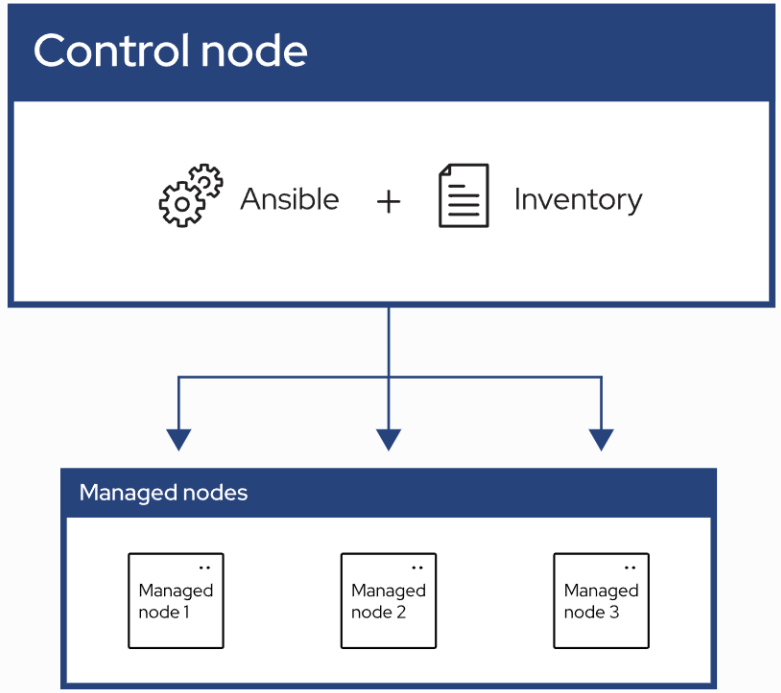
Grafana TCP 3000

WebUI – server:3000

Ansible

- Automation using "playbooks" written in YAML format
- Agent-less architecture
  - No installation on the managed nodes
- Idempotency
  - Does not change/write if target is already in state described by playbook

- Control node
  - System where Ansible is installed and runs the playbooks
- Inventory
  - List of managed nodes where the playbook will affect
- Managed nodes
  - Remote systems that are the targets of your playbooks



Pod numbers

- Students get pod numbers ranging from 11 and up
- Your WLC will have last octed equal to your assigned Pod#. Ubuntu will have Pod# +20

Pod#	WLC IP	Ubuntu IP	Name
11	192.168.10.11	192.168.10.31	Tauni
12	192.168.10.12	192.168.10.32	Samuel
13	192.168.10.13	192.168.10.33	Payman
14	192.168.10.14	192.168.10.34	Hans
15	192.168.10.15	192.168.10.35	Michal
16	192.168.10.16	192.168.10.36	Stephen
17	192.168.10.17	192.168.10.37	Sigurd
18	192.168.10.18	192.168.10.38	Konrad
19	192.168.10.19	192.168.10.39	Ed
20	192.168.10.20	192.168.10.40	Yifan
21	192.168.10.21	192.168.10.41	Nick
22	192.168.10.22	192.168.10.42	Marek
23	192.168.10.23	192.168.10.43	Martin
24	192.168.10.24	192.168.10.44	Igor
25	192.168.10.25	192.168.10.45	Piotr
26	192.168.10.26	192.168.10.46	Tór
29	192.168.10.29	192.168.10.49	Kjetil
30	192.168.10.30	192.168.10.50	Andreas

Feedback on pre-lab tasks?

- We really need feedback to make this deep dive better ☺
- Pre-lab or anything else!