World Wide Technology

meetup

Data manipulation techniques for configuration management using Ansible

February 2020

Joel W. King Engineering and Innovations
Global Solutions Development

title | abstract

For Reference

Title:
Data manipulation techniques for configuration management using Ansible

What is the most important thing that people will learn from your presentation?:
This talk explores techniques and best practices for ingesting, manipulating and storing configuration management data for managing multi-cloud infrastructure deployments using Ansible.

Abstract:
Ansible is widely adopted as a configuration management tool for both on-premise infrastructure and multi-cloud deployments. Most learning tracks focus on procedural programming concepts, learning playbook syntax. This talk focuses on techniques to ingest, manipulate and optimize configuration management data to drive the process. We examine techniques to create data sinks for audit and input to downstream workflows. Also highlighted is the application of relational, NoSQL and graph databases as well as sequential files used for configuration management.

whoami

For Reference

SlideShare

www.slideshare.net/joelwking/

World Wide Technology

https://www.wwt.com/

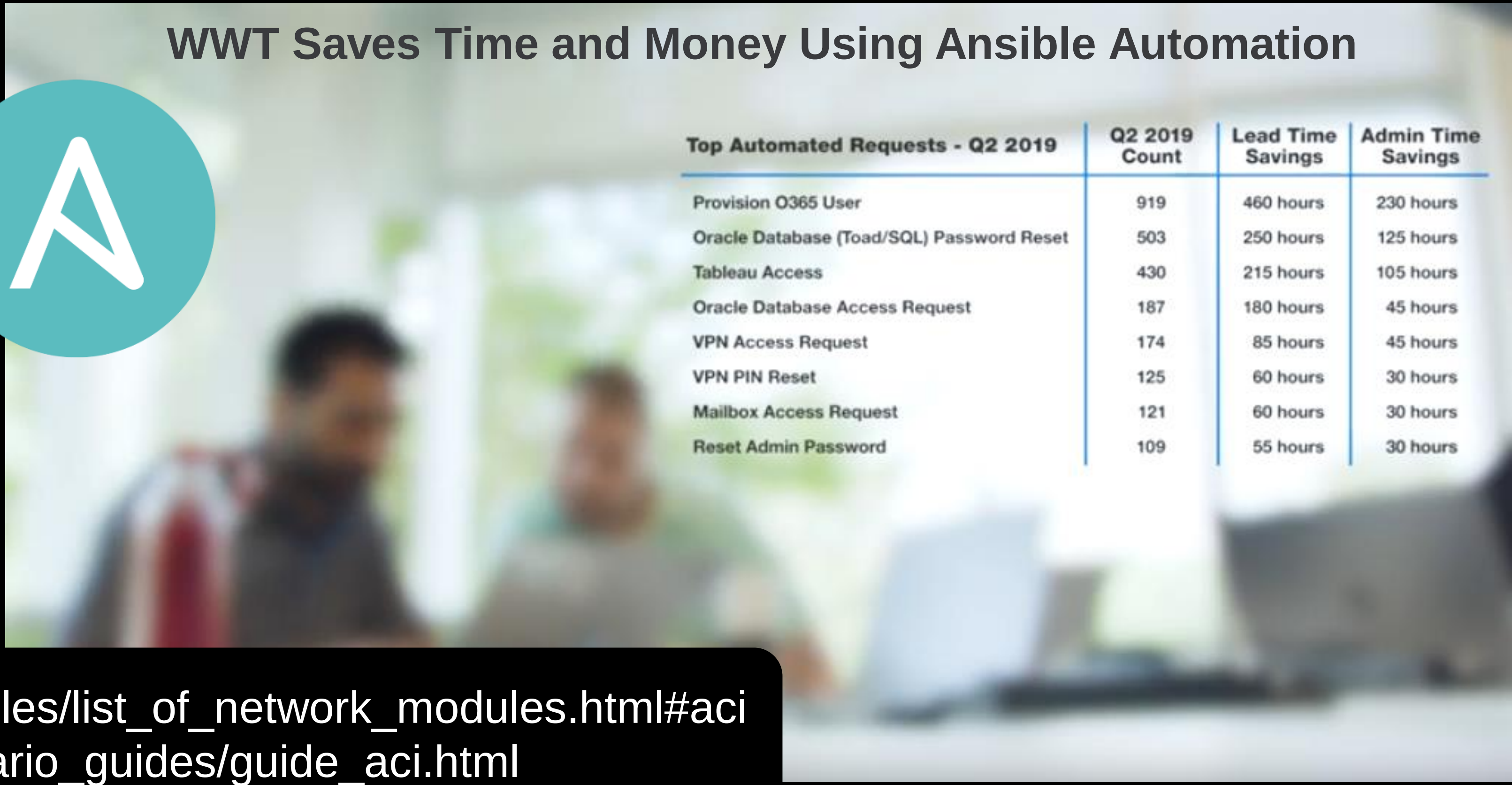
@joel_w_king

@joelwking

@programmablenetworks/

- ✓ Low Barrier to Entry
- ✓ Highly Extensible
- ✓ Broad Industry adoption for Configuration Management

- ✓ ACI Modules
  100 for ACI and MSO



WWT Saves Time and Money Using Ansible Automation

Top Automated Requests - Q2 2019	Q2 2019 Count	Lead Time Savings	Admin Time Savings
Provision O365 User	919	460 hours	230 hours
Oracle Database (Toad/SQL) Password Reset	503	250 hours	125 hours
Tableau Access	430	215 hours	105 hours
Oracle Database Access Request	187	180 hours	45 hours
VPN Access Request	174	85 hours	45 hours
VPN PIN Reset	125	60 hours	30 hours
Mailbox Access Request	121	60 hours	30 hours
Reset Admin Password	109	55 hours	30 hours

docs.ansible.com/ansible/latest/modules/list_of_network_modules.html#aci
docs.ansible.com/ansible/latest/scenario_guides/guide_aci.html

https://youtu.be/3vuPRoyOIFo

agenda

AUTOMATION: BASICS: variables and facts

AUTOMATION: ADVANCED FEATURES: plugins | facts modules

USE CASES

AUTOMATION: PRIVATE CLOUD : Cisco ACI

AUTOMATION: SECURITY AND COMPLIANCE

SHARE YOUR

KNOWLEDGE
EXPERIENCE

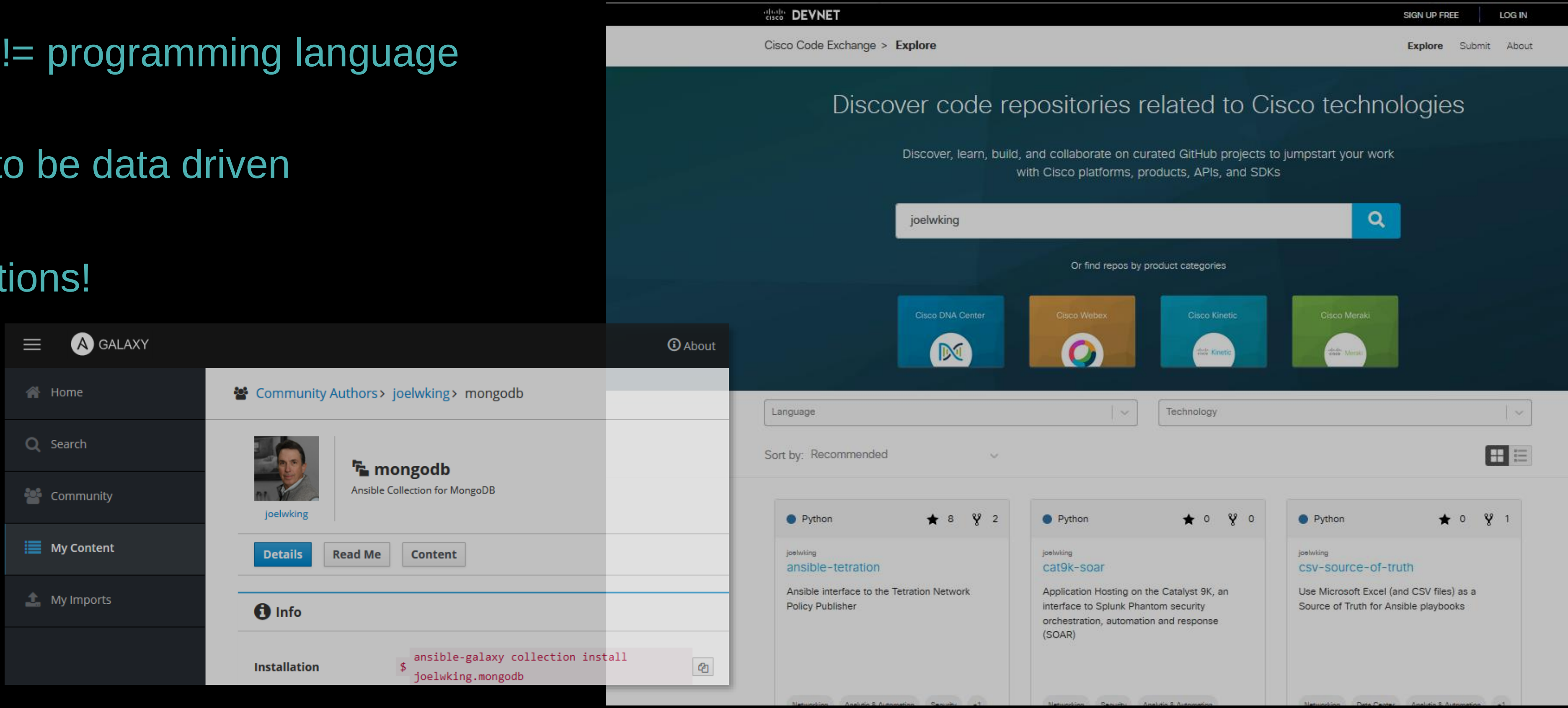
level set | caveats

Background and use cases are typically network and security automation focused.

Ansible Playbooks != programming language

Design playbooks to be data driven

Use Ansible Collections!



Discover code repositories related to Cisco technologies

Discover, learn, build, and collaborate on curated GitHub projects to jumpstart your work with Cisco platforms, products, APIs, and SDKs

joelwking

Or find repos by product categories

Sort by: Recommended

Python ★ 8 2
joelwking
ansible-tetration
Ansible interface to the Tetration Network Policy Publisher

Python ★ 0 0
joelwking
cat9k-soar
Application Hosting on the Catalyst 9K, an interface to Splunk Phantom security orchestration, automation and response (SOAR)

Python ★ 0 1
joelwking
csv-source-of-truth
Use Microsoft Excel (and CSV files) as a Source of Truth for Ansible playbooks

Community Authors › joelwking › mongodb

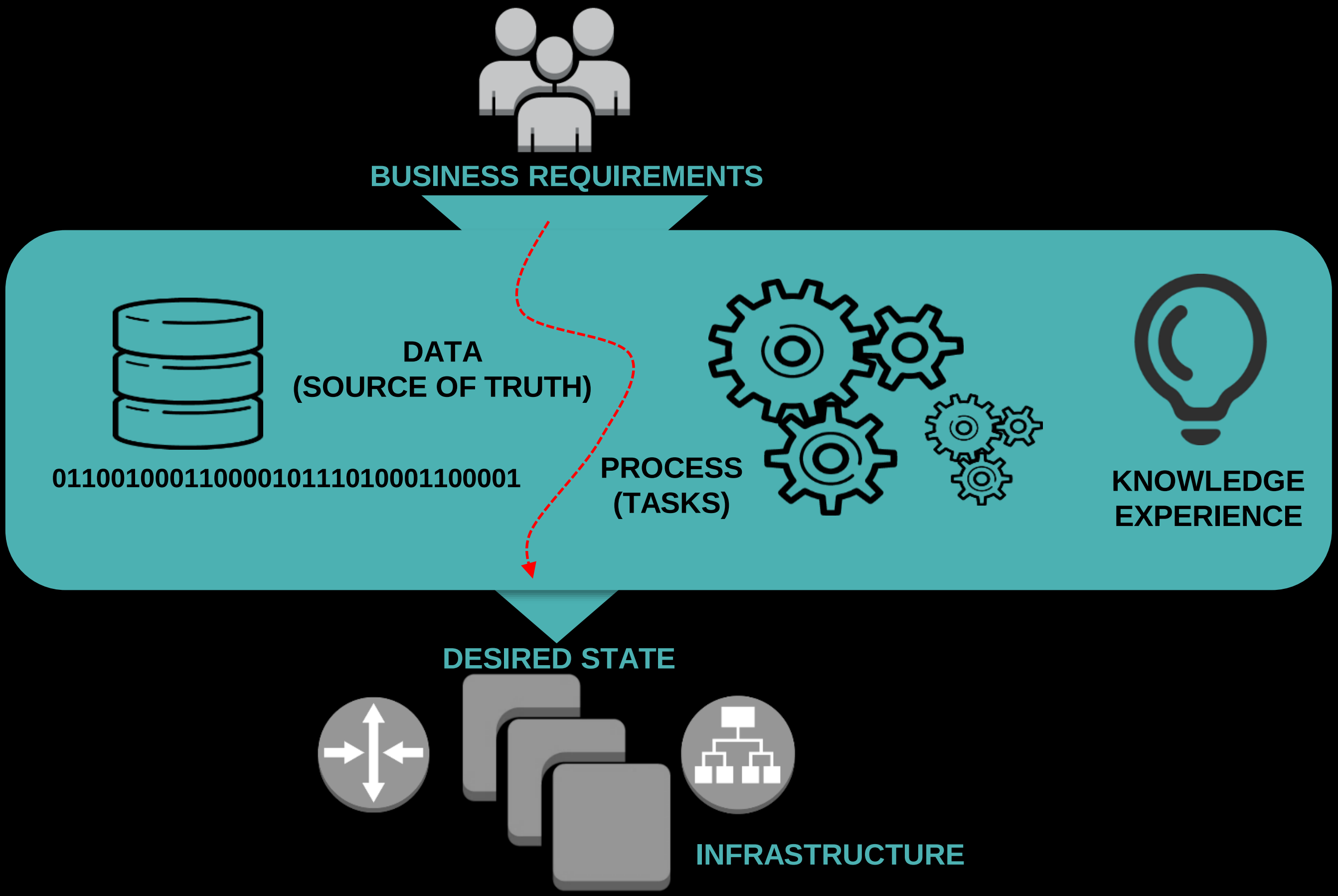
mongodb
Ansible Collection for MongoDB

joelwking

Installation

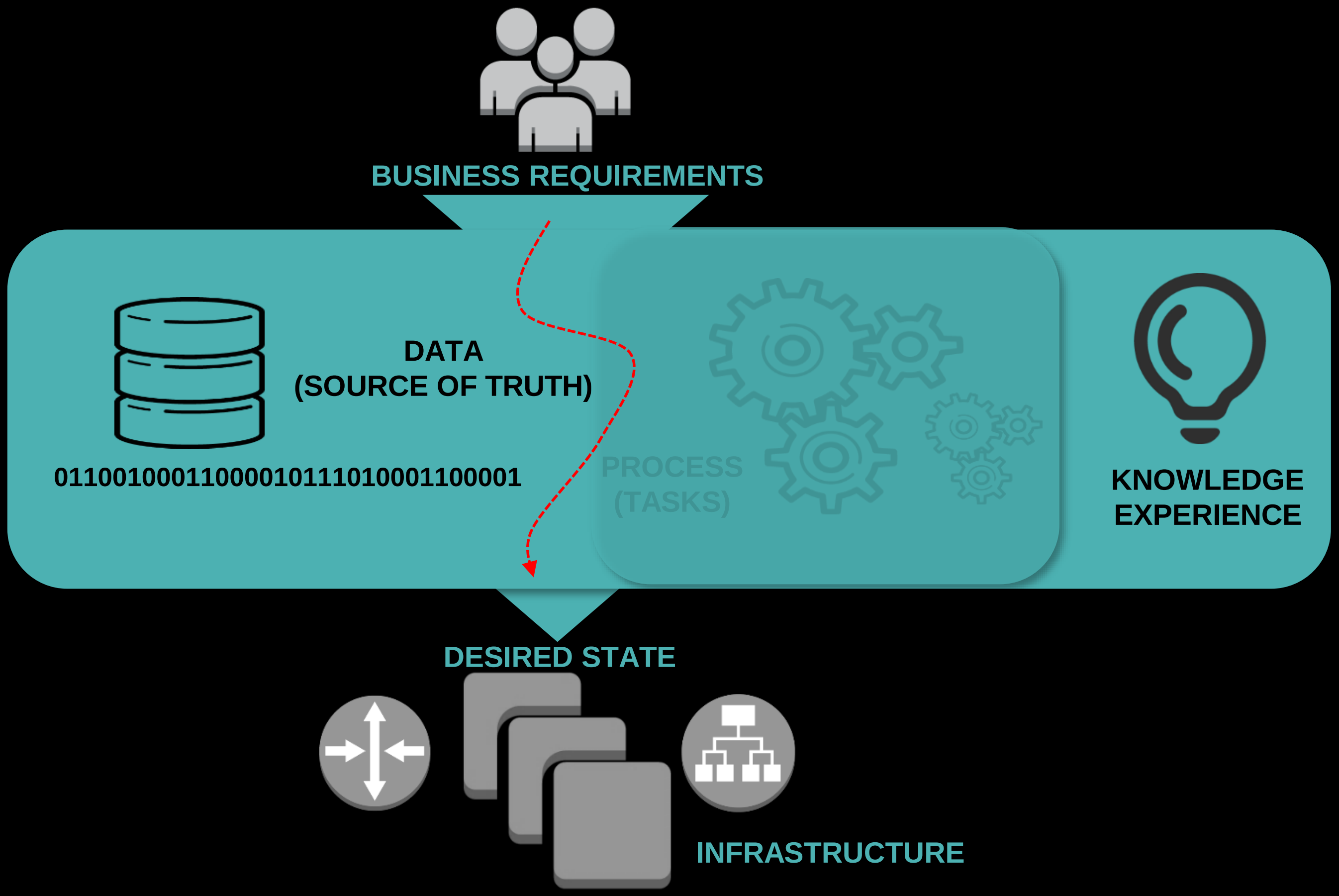
```
$ ansible-galaxy collection install joelwking.mongodb
```

https://developer.cisco.com/codeexchange/explore#search=joelwking

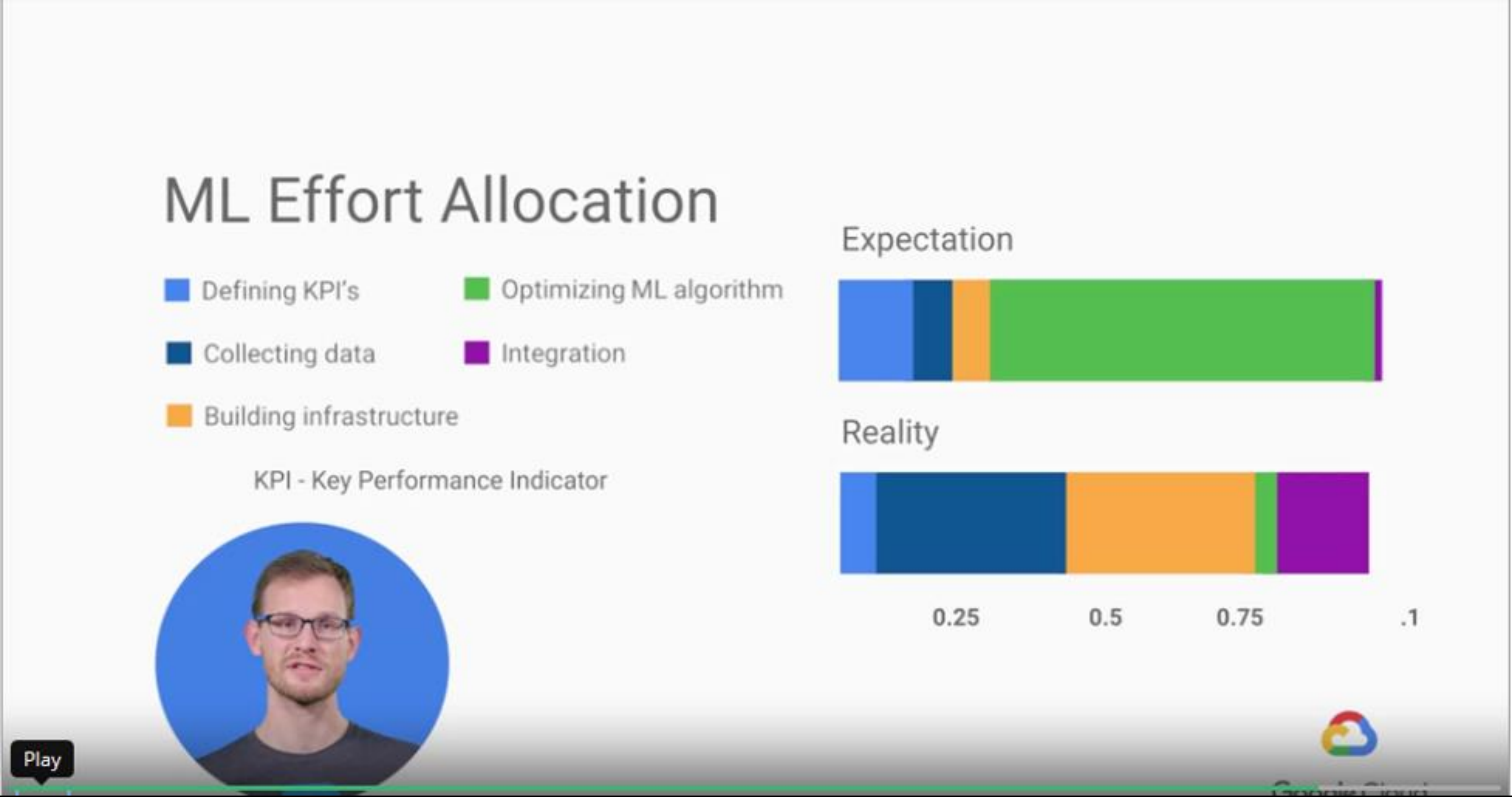

BUSINESS REQUIREMENTS
DATA
(SOURCE OF TRUTH)
01100100011000010111010001100001
PROCESS
(TASKS)
KNOWLEDGE
EXPERIENCE
DESIRED STATE
INFRASTRUCTURE

How Google does Machine Learning

… It's all about data



https://www.coursera.org/learn/google-machine-learning/



ML Effort Allocation

- 🟦 Defining KPI's
- 🟦 Collecting data
- 🟧 Building infrastructure
- 🟩 Optimizing ML algorithm
- 🟪 Integration

KPI - Key Performance Indicator

AUTOMATION: BASICS

#SiliconValleyinSTL

configuration data

HOSTS

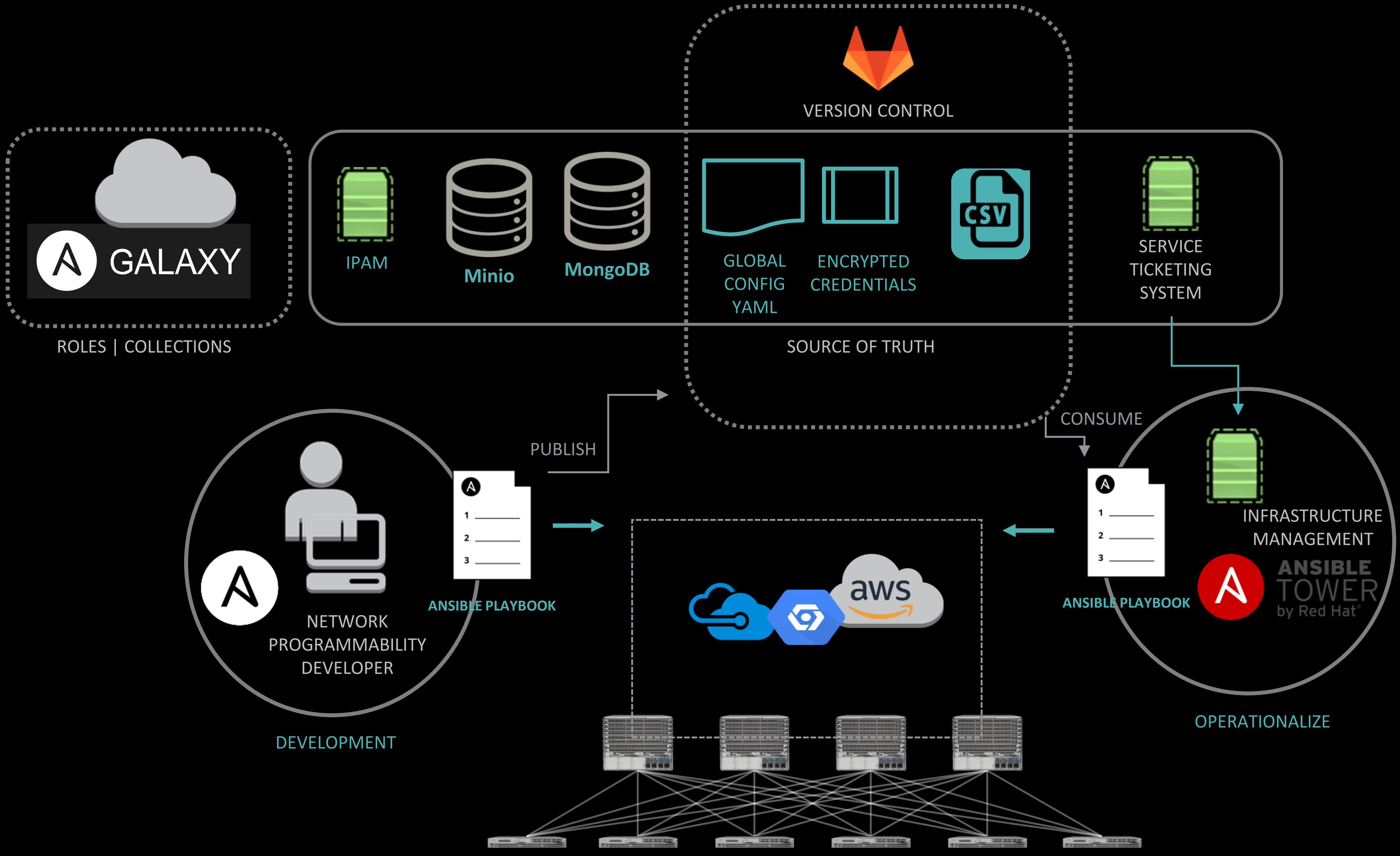
NETWORK DEVICES

{ API }

Variables can be defined in a bewildering variety of places in an Ansible project.(*)

Ansible facts are variables that are automatically discovered by Ansible from a managed host. Facts are pulled by the setup module.(*)

(*) Automation with Ansible -Student Workbook



GALAXY

ROLES | COLLECTIONS

VERSION CONTROL

IPAM

Minio

MongoDB

GLOBAL CONFIG YAML

ENCRYPTED CREDENTIALS

CSV

SERVICE TICKETING SYSTEM

SOURCE OF TRUTH

PUBLISH

CONSUME

ANSIBLE PLAYBOOK

NETWORK PROGRAMMABILITY DEVELOPER

DEVELOPMENT

aws

INFRASTRUCTURE MANAGEMENT

ANSIBLE TOWER by Red Hat

ANSIBLE PLAYBOOK

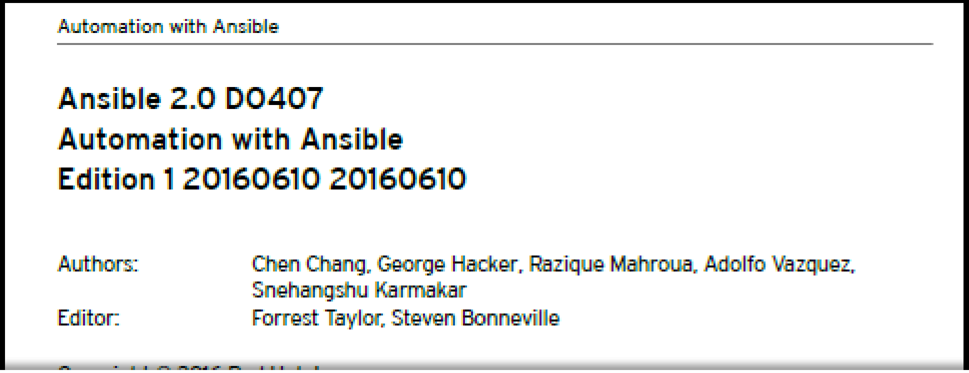
OPERATIONALIZE

variable precedence

least

command line values (eg “-u user”)
role defaults [1]
inventory file or script group vars [2]
inventory group_vars/all [3]
playbook group_vars/all [3]
inventory group_vars/* [3]
playbook group_vars/* [3]
inventory file or script host vars [2]
inventory host_vars/* [3]
playbook host_vars/* [3]
host facts / cached set_facts [4]
play vars
play vars_prompt
play vars_files
role vars (defined in role/vars/main.yml)
block vars (only for tasks in block)
task vars (only for the task)
include_vars
set_facts / registered vars
role (and include_role) params
include params
extra vars (always win precedence)

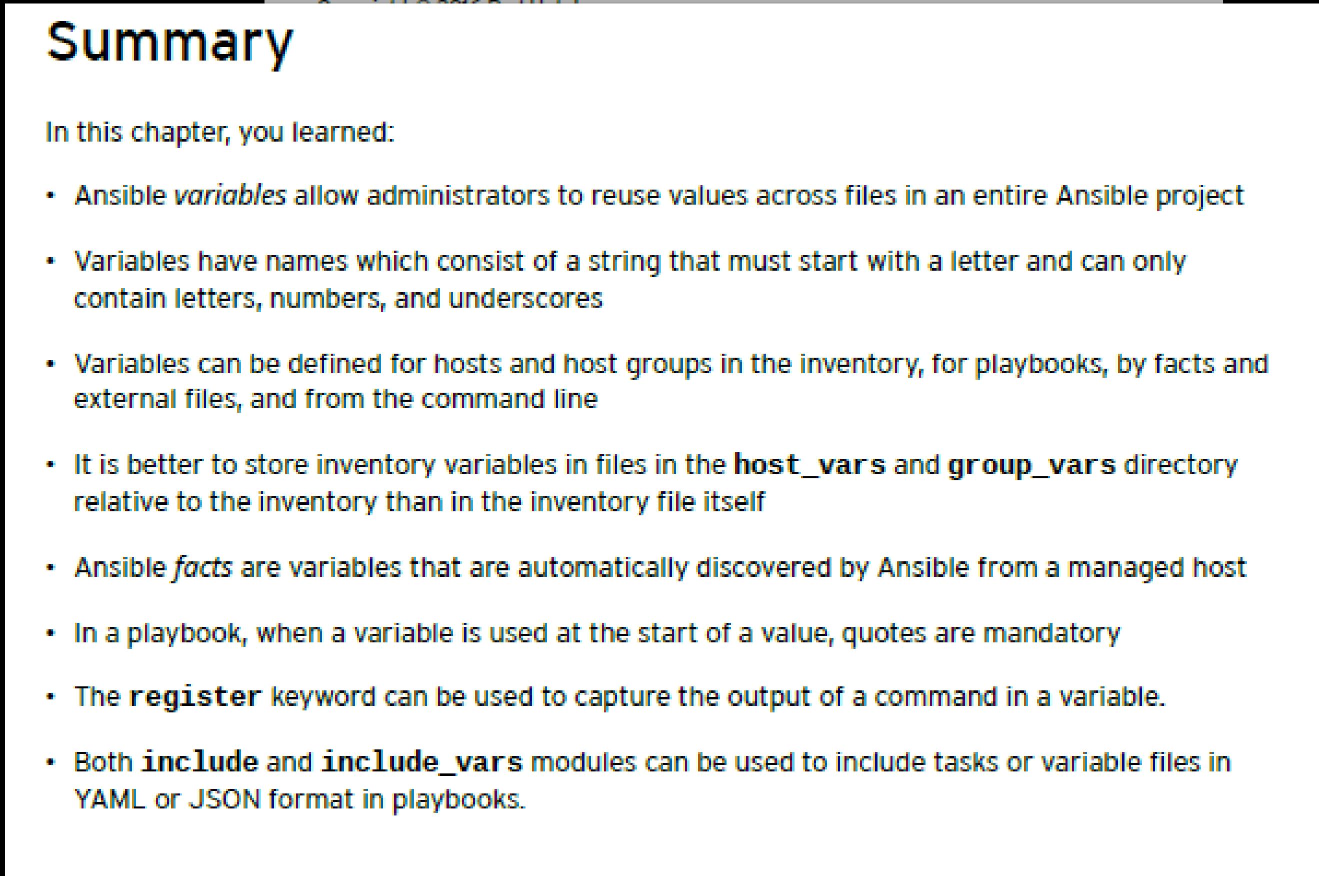
most



Automation with Ansible

Ansible 2.0 DO407
Automation with Ansible
Edition 1 20160610 20160610

Authors:	Chen Chang, George Hacker, Razique Mahroua, Adolfo Vazquez, Snehangshu Karmakar
Editor:	Forrest Taylor, Steven Bonneville



Summary

In this chapter, you learned:

- Ansible *variables* allow administrators to reuse values across files in an entire Ansible project
- Variables have names which consist of a string that must start with a letter and can only contain letters, numbers, and underscores
- Variables can be defined for hosts and host groups in the inventory, for playbooks, by facts and external files, and from the command line
- It is better to store inventory variables in files in the **host_vars** and **group_vars** directory relative to the inventory than in the inventory file itself
- Ansible *facts* are variables that are automatically discovered by Ansible from a managed host
- In a playbook, when a variable is used at the start of a value, quotes are mandatory
- The **register** keyword can be used to capture the output of a command in a variable.
- Both **include** and **include_vars** modules can be used to include tasks or variable files in YAML or JSON format in playbooks.

https://docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html#variable-precedence-where-should-i-put-a-variable

yaml to python

```
#!/usr/bin/ansible-playbook
---
#
#
- name: Data Manipulation | Meetup | Ansible Durham | S3
  hosts: s3.amazonaws.com
  connection: local
  gather_facts: '{{ facts | default("no") }}'

  module_defaults:
    aws_s3:
      aws_access_key: '{{ access.key }}'
      aws_secret_key: '{{ secret.key }}'
      bucket: '{{ bucket }}'

  vars:
    input:
      - name: '/usr/bin/ansible'
        state: put
        meta_data: 'foo=bar'
      - name: '{{ playbook_dir }}/files/csv123.csv'
        state: put
        meta_data: 'foo=bar'
```

```
administrator@flint:~/ansible/playbooks$ python
Python 2.7.12 (default, Oct  8 2019, 14:14:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import yaml
>>> input = '''
...      - name: '/usr/bin/ansible'
...        state: put
...        meta_data: 'foo=bar'
...      - name: '{{ playbook_dir }}/files/csv123.csv'
...        state: put
...        meta_data: 'foo=bar'
... '''
>>> data = yaml.load(input)
>>> type(data)
<type 'list'>
>>> type(data[0])
<type 'dict'>
>>> data[0]
{'state': 'put', 'meta_data': 'foo=bar', 'name': '/usr/bin/ansible'}
>>>
```

best practices: variables

Proper variable naming can make plays more readable and avoid variable name conflicts(*)

Use descriptive, unique human-meaningful variable names(*)

Prefix variables with it’s “owner” such as a role name or package(*)

```
vars:
       mongodb_database:  "{{ database }}"
       mongodb_collection: "{{ collection }}"
       mongodb_document: "{{ filename }}"



 collection: "{{ mongodb_collection }}"
```

```
vars:
  dns:
    ipv6:
       - '2001:4860:4860::8888'
       - '2001:4860:4860::8844'



    dns6: "{{ dns.ipv6 }}"
```

(*) ANSIBLE BEST PRACTICES: THE ESSENTIALS – Kyle Benson

data_manipulation_debug.yml

Difference between?

-e 'desc=this is a test’

and

-e 'desc="this is a test"'

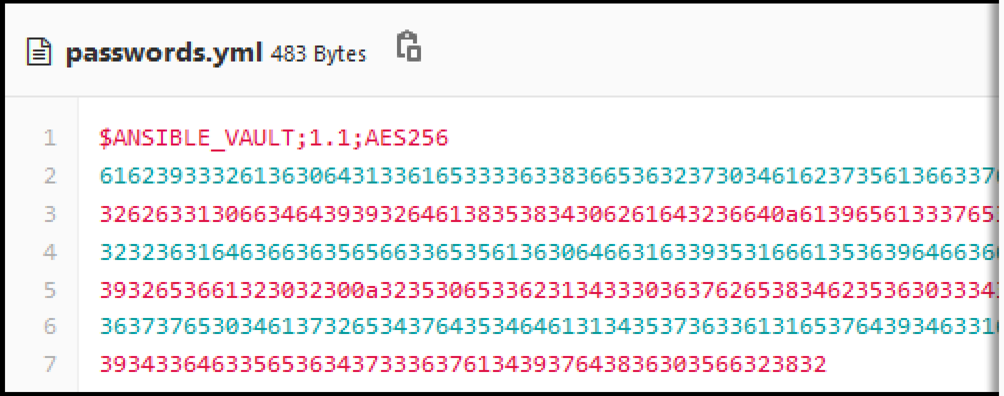
Why might you run gather facts on a local host?

Where is variable ‘imdata’ defined?

Using a dictionary rather than a scalar variable

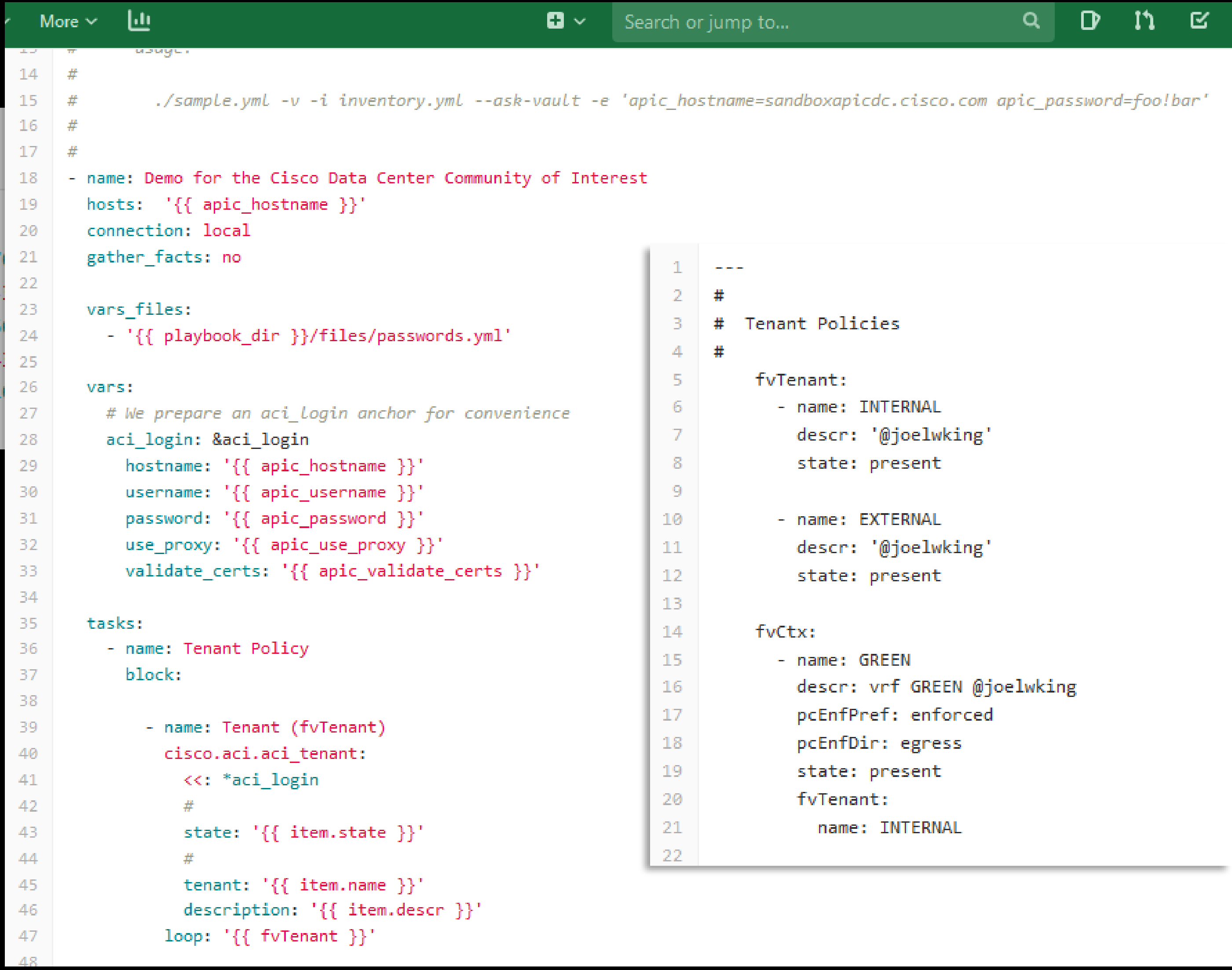
```
- name: Data Manipulation | Meetup | Ansible Durham
  hosts: sandboxapicdc.cisco.com
  connection: local
  gather_facts: '{{ facts | default("no") }}'

  vars:
    foo:
      name: bar
      description: '{{ desc | default("This is a test") }}'

  vars_files:
    - '{{ playbook_dir }}/files/LTM_vip_v6-3.json'

  tasks:
    - name: Hostvars | run with or without gathering facts
      debug:
        var: hostvars[inventory_hostname]

    - name: Variable from vars_file specified
      debug:
        var: imdata

    - name: Variable defined in playbook
      debug:
        msg: '{{ foo.description }}'
```

let data drive the playbook



passwords.yml 483 Bytes

```
$ANSIBLE_VAULT;1.1;AES256
616239333261363064313361653333633836653632373034616237356136633
32626331306634643939326461383538343062616432366 40a6139656133376 5
32323631646366363565663365356136306466316339353166613536396466363
39326536613230323 00a3235306533623134333036376265383462353630333 4
36373765303461373265343764353464613134353736336131653764393463316
39343364633565363437333637613439376438363035663 23832
```



```
#       usage:
#
#       ./sample.yml -v -i inventory.yml --ask-vault -e 'apic_hostname=sandboxapicdc.cisco.com apic_password=foo!bar'
#
#
- name: Demo for the Cisco Data Center Community of Interest
  hosts:  '{{ apic_hostname }}'
  connection: local
  gather_facts: no

  vars_files:
    - '{{ playbook_dir }}/files/passwords.yml'

  vars:
    # We prepare an aci_login anchor for convenience
    aci_login: &aci_login
      hostname: '{{ apic_hostname }}'
      username: '{{ apic_username }}'
      password: '{{ apic_password }}'
      use_proxy: '{{ apic_use_proxy }}'
      validate_certs: '{{ apic_validate_certs }}'

  tasks:
    - name: Tenant Policy
      block:

        - name: Tenant (fvTenant)
          cisco.aci.aci_tenant:
            <<: *aci_login
            #
            state: '{{ item.state }}'
            #
            tenant: '{{ item.name }}'
            description: '{{ item.descr }}'
          loop: '{{ fvTenant }}'
```

```
---
#
#  Tenant Policies
#

    fvTenant:
      - name: INTERNAL
        descr: '@joelwking'
        state: present

      - name: EXTERNAL
        descr: '@joelwking'
        state: present

    fvCtx:
      - name: GREEN
        descr: vrf GREEN @joelwking
        pcEnfPref: enforced
        pcEnfDir: egress
        state: present
        fvTenant:
          name: INTERNAL

```

inventory.yml

```
---
all:
  children:
    APIC:
      hosts: sandboxapicdc.cisco.com:
          apic_hostname: sandboxapicdc.cisco.com
          apic_username: admin
          apic_use_proxy: no
          apic_validate_certs: no
          apic_password: foo!bar
```

https://gitlab.com/joelwking/cisco_dc_community_of_interest/-/blob/master/demos/engine/playbooks/sample.yml

SHARE YOUR

KNOWLEDGE
EXPERIENCE

AUTOMATION: ADVANCED TOPICS

#SiliconValleyinSTL

Modules have modes to ‘put’ or ‘get’

aws_s3

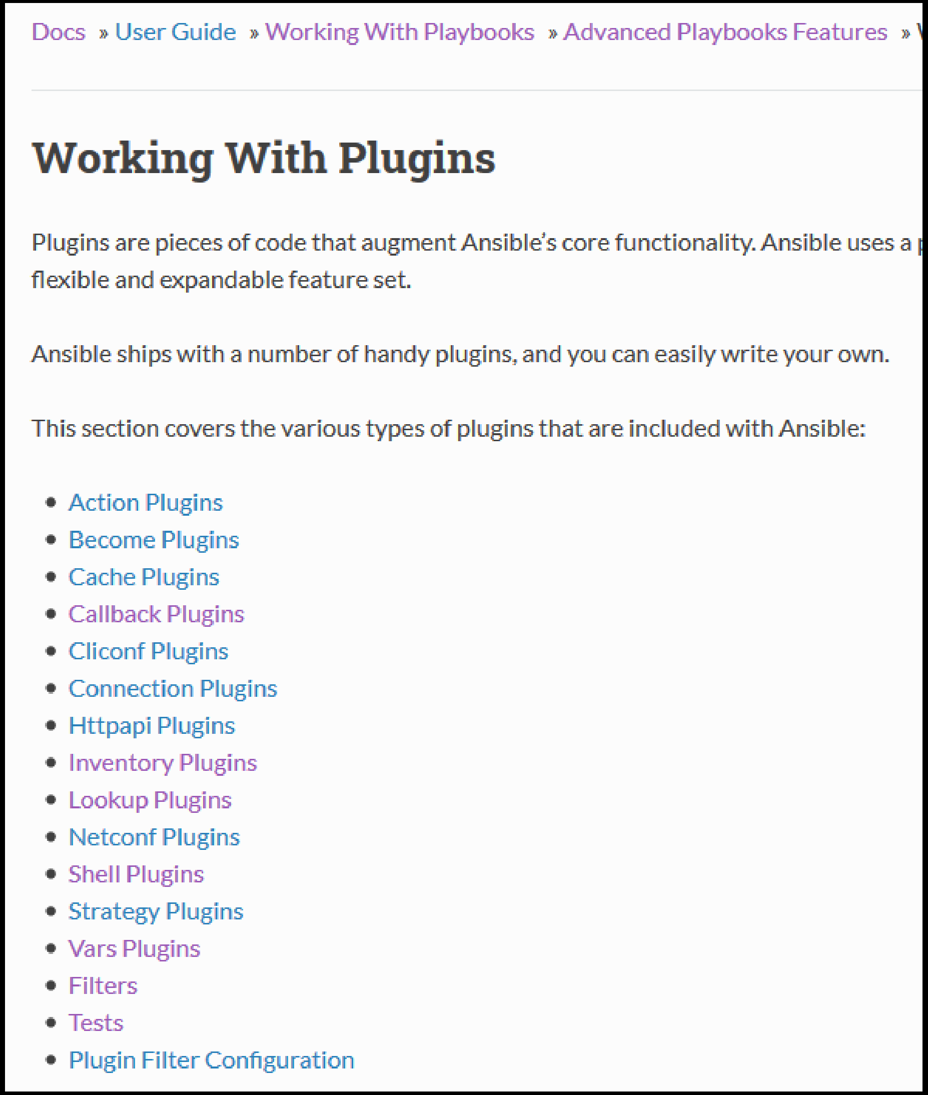
Modules are used as information gatherers

_info
_facts

Lookup plugins allow Ansible to access data from outside sources.

Lookups are an integral part of loops

Filters in Ansible… are used for transforming data …



Docs » User Guide » Working With Playbooks » Advanced Playbooks Features » V

Working With Plugins

Plugins are pieces of code that augment Ansible’s core functionality. Ansible uses a p flexible and expandable feature set.

Ansible ships with a number of handy plugins, and you can easily write your own.

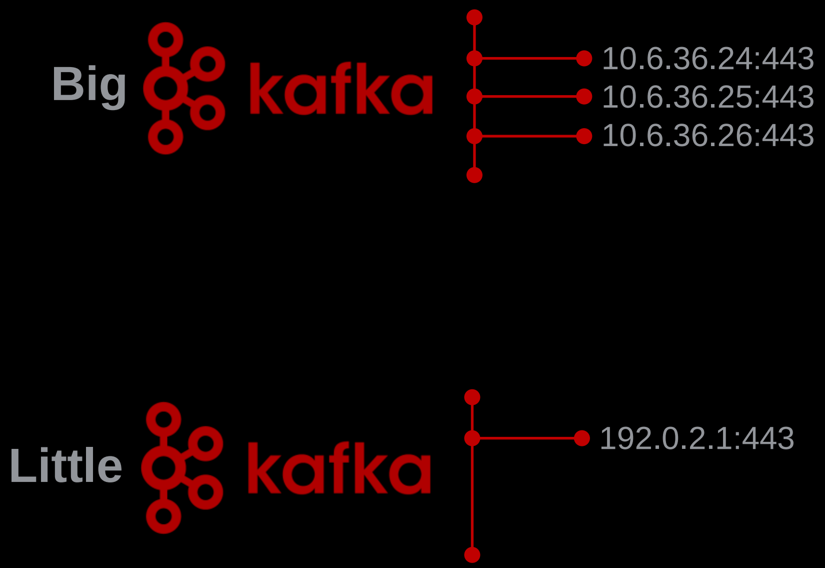
This section covers the various types of plugins that are included with Ansible:

- Action Plugins
- Become Plugins
- Cache Plugins
- Callback Plugins
- Cliconf Plugins
- Connection Plugins
- Httpapi Plugins
- Inventory Plugins
- Lookup Plugins
- Netconf Plugins
- Shell Plugins
- Strategy Plugins
- Vars Plugins
- Filters
- Tests
- Plugin Filter Configuration

https://docs.ansible.com/ansible/latest/plugins/plugins.html

lookup plugins

FORM FACTORS



return random element from list

filter returning boolean

```
#!/usr/bin/ansible-playbook
---
#
#      Copyright (c) 2019 World Wide Technology, Inc.
#      All rights reserved.
#
#      author: Joel W. King,  World Wide Technology
#
- name: Msg broker for big or little Tetration
  hosts: localhost
  gather_facts: yes
  connection: local

  vars:
    verify_broker: 'no'
    big_tetration: '10.6.36.24:443,10.6.36.25:443,10.6.36.26:443'
    little_tetration: '192.0.2.1:443'

  tasks:
    - debug:
        msg: 'Big Tetration have three Kafka brokers, pick one: {{ item }}'
      with_random_choice: '{{ big_tetration.split(",") }}'

    - debug:
        msg: 'Little Tetration only has one Kafka broker: {{ item }}'
      with_random_choice: '{{ little_tetration.split(",") }}'

    - set_fact:
        kafka_broker: '{{ item }}'
      with_random_choice: '{{ big_tetration.split(",") }}'

    - name: Optionally verify reachability to the broker
      shell: 'openssl s_client  -connect {{ kafka_broker }}  -tls1 '
      ignore_errors: False
      register: openssl
      when:  verify_broker|bool
```

using a dictionary to translate data values

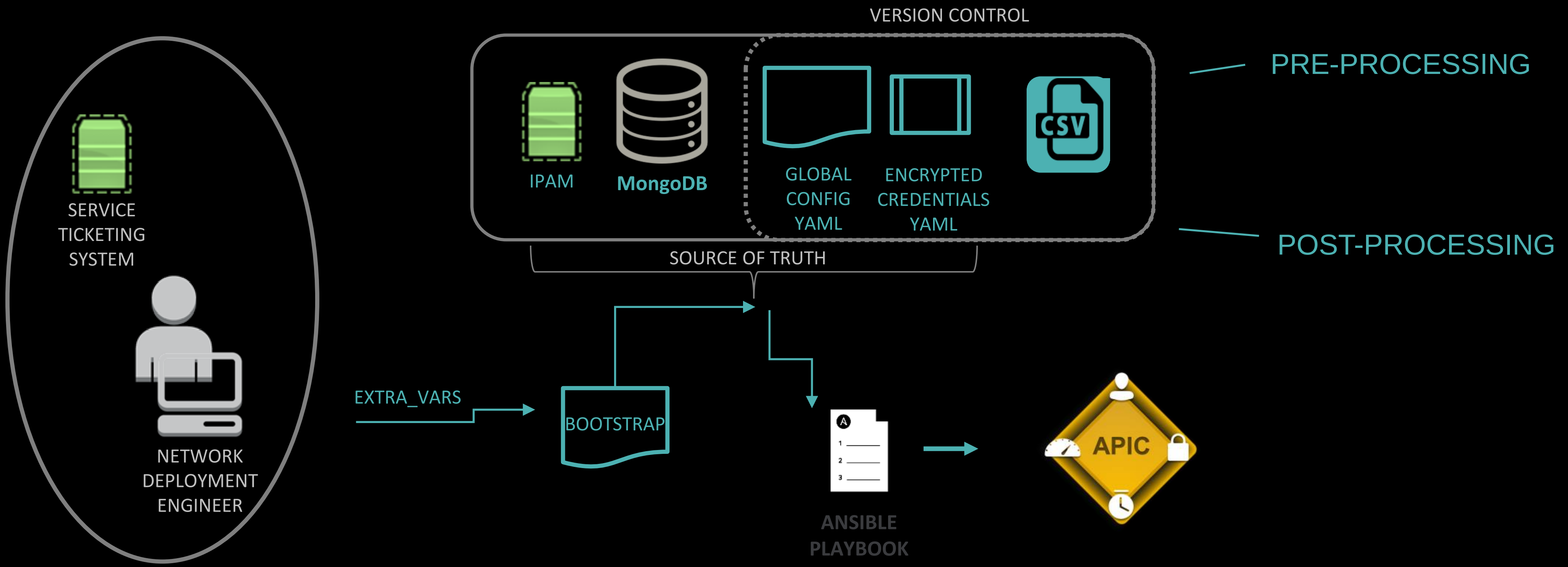
```
acl_action: {"ALLOW": "permit", "DROP": "deny"}        # translator for action verbs
```

```
- name: Configure firewall access-list
 debug:
   msg: "access-list {{ acl_name }} line 1 extended {{ acl_action[item.action] }} {{item.ip_protocol }}"
 loop: "{{ tnp.ansible_facts.intents }}"
```

Policy engine uses different keywords than the firewall where the policy is being applied

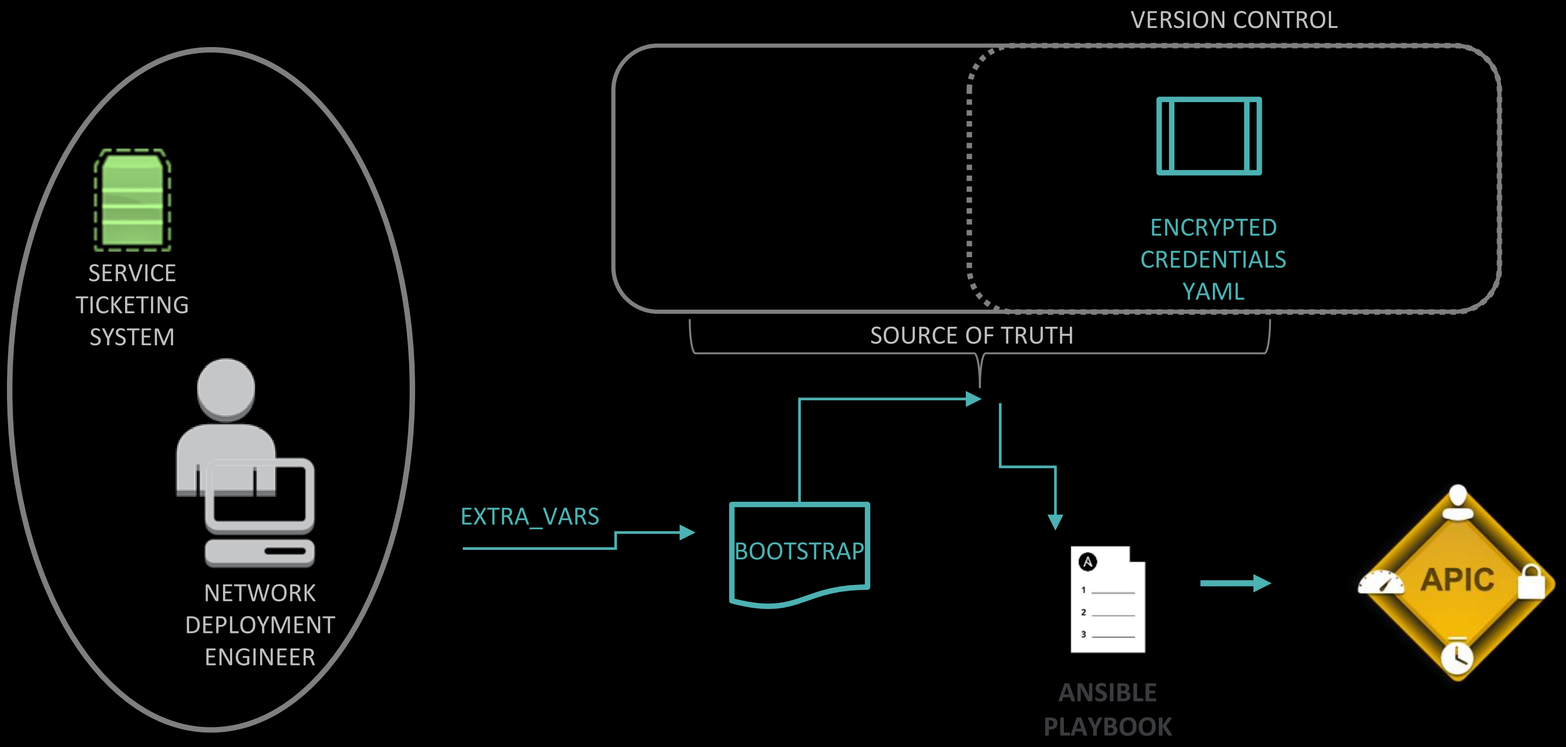
https://gitlab.com/tetration-network-policy-publisher/policy-stream-12-pub-vrf/-/blob/master/view_network_policy_decrypt.yml

use roles for ingesting data



https://github.wwt.com/it-automation-aci/ansible-aci-include-data

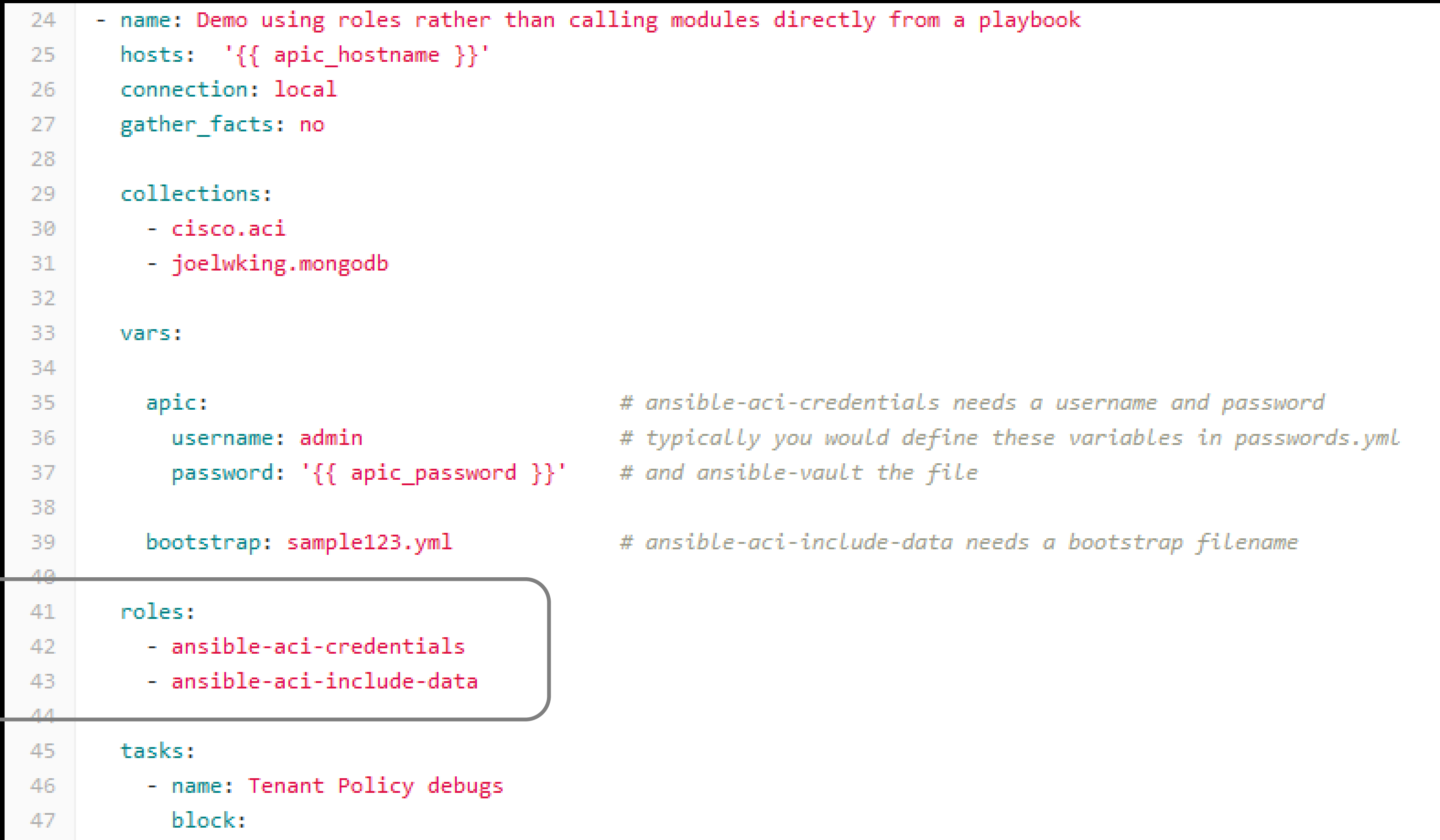
use roles for managing credentials



https://github.wwt.com/it-automation-aci/ansible-aci-credentials

playbooks for ACI

./sample.yml -v -i inventory.yml --e 'bootstrap=sample456

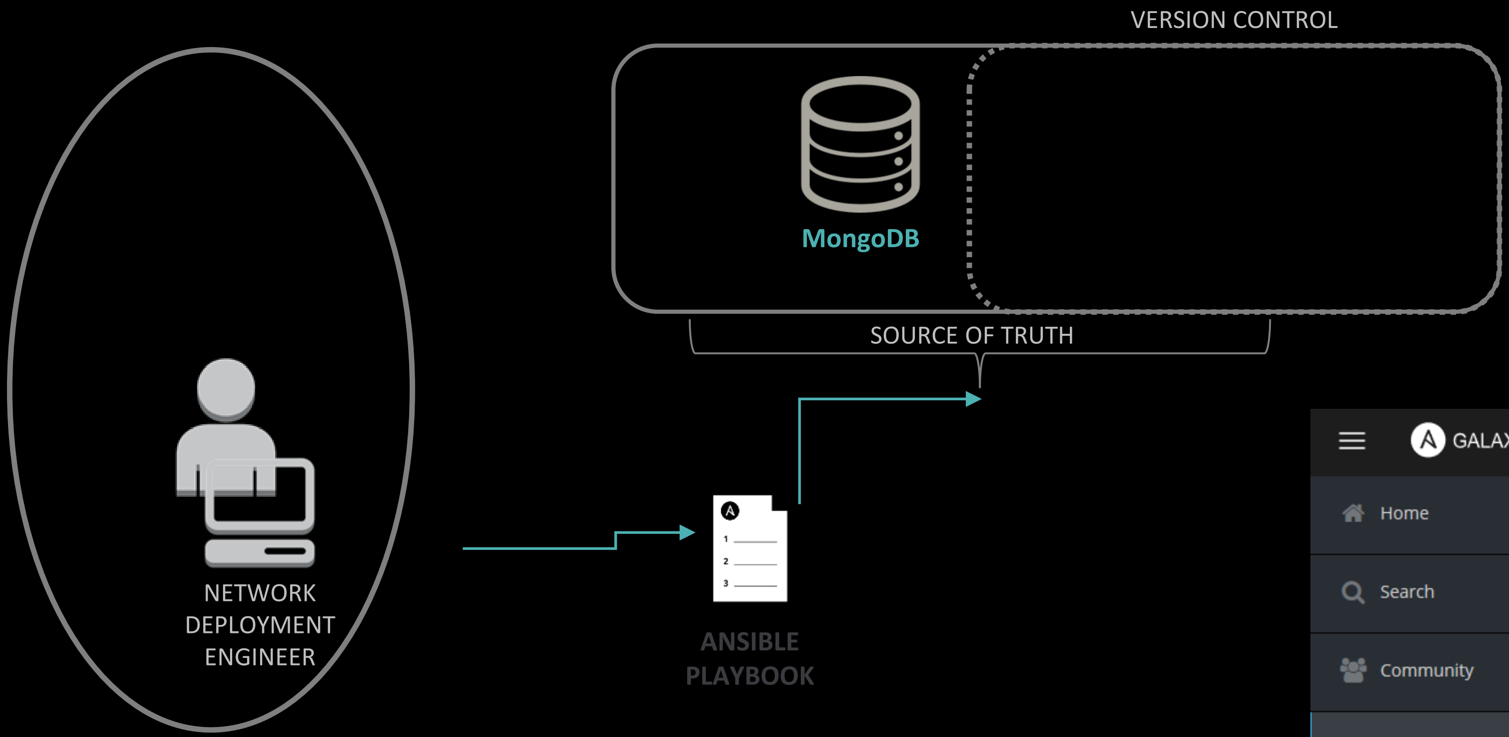


```
- name: Demo using roles rather than calling modules directly from a playbook
  hosts:  '{{ apic_hostname }}'
  connection: local
  gather_facts: no

  collections:
    - cisco.aci
    - joelwking.mongodb

  vars:

    apic:                                # ansible-aci-credentials needs a username and password
      username: admin                    # typically you would define these variables in passwords.yml
      password: '{{ apic_password }}'    # and ansible-vault the file

    bootstrap: sample123.yml             # ansible-aci-include-data needs a bootstrap filename

  roles:
    - ansible-aci-credentials
    - ansible-aci-include-data

  tasks:
    - name: Tenant Policy debugs
      block:
```

All playbooks call these roles for managing credentials and including data

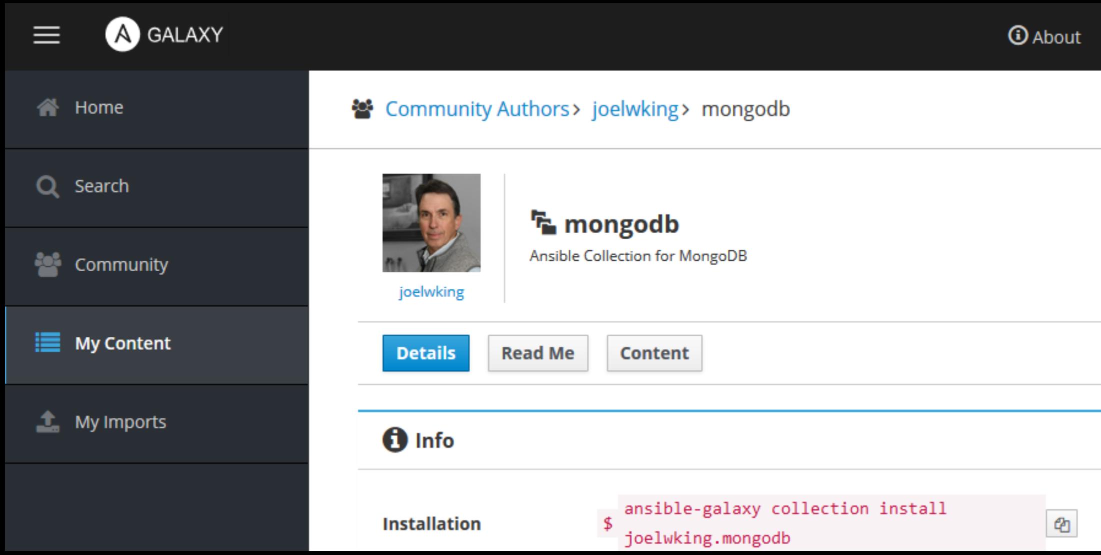
https://gitlab.com/joelwking/cisco_dc_community_of_interest/-/blob/master/demos/engine/playbooks/sample_wwt_roles.yml

using MongoDB as a data store



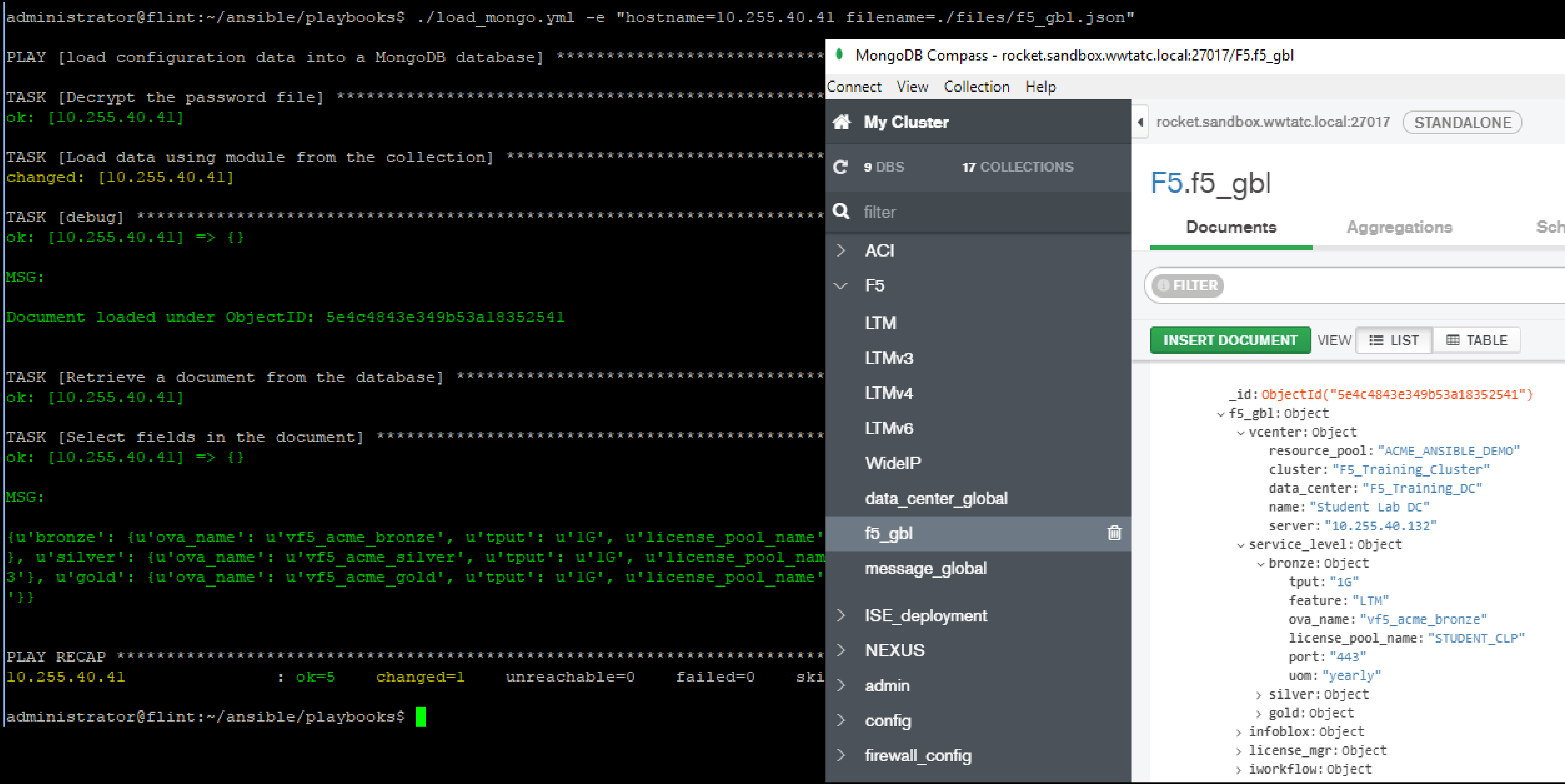
MongoDB is an open-source document / NoSQL database.

Implemented as a Docker Container



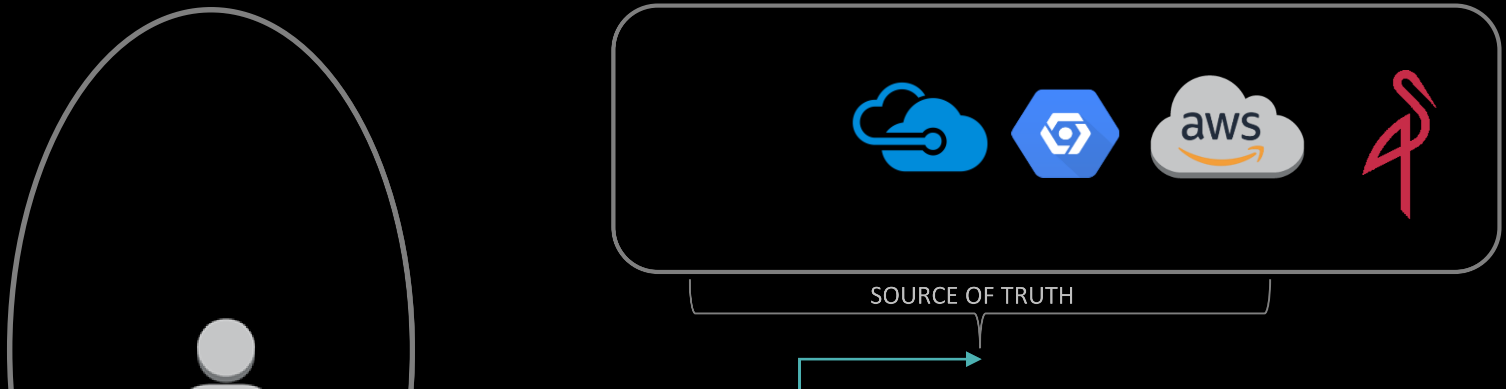
Installation

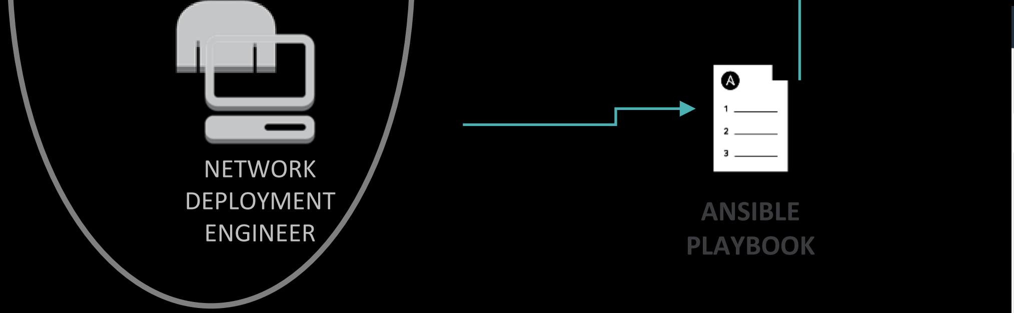
```
$ ansible-galaxy collection install
joelwking.mongodb
```

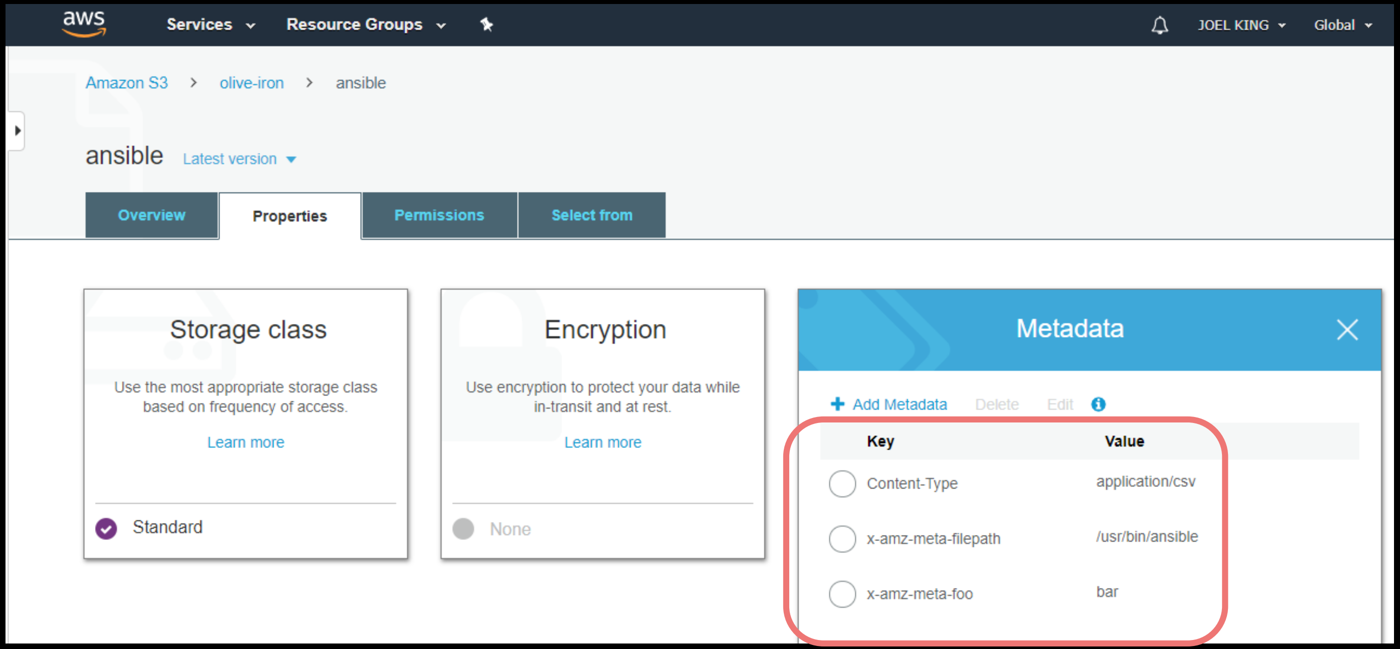


using object storage as a data store

Amazon S3 or Amazon Simple Storage Service is a service offered by Amazon Web Services that provides object storage through a web service interface.







Amazon S3 > olive-iron > ansible

ansible Latest version

Storage class

Use the most appropriate storage class based on frequency of access.

Learn more

● Standard

Encryption

Use encryption to protect your data while in-transit and at rest.

Learn more

● None

Metadata

	Key	Value
○	Content-Type	application/csv
○	x-amz-meta-filepath	/usr/bin/ansible
○	x-amz-meta-foo	bar

For Reference

```yaml
- name: Data Manipulation | Meetup | Ansible Durham | S3
  hosts: s3.amazonaws.com
  connection: local
  gather_facts: '{{ facts | default("no") }}'

  module_defaults:
    aws_s3:
      aws_access_key: '{{ access.key }}'
      aws_secret_key: '{{ secret.key }}'
      bucket: '{{ bucket }}'

  vars:
    keys: '/home/administrator/amazon_web_service/access_keys/s3.yml'
    input:
      - name: '/usr/bin/ansible'
        state: put
        meta_data: 'foo=bar'
      - name: '{{ playbook_dir }}/files/sample_csv_file.csv'
        state: put
        meta_data: 'foo=bar'

  vars_files:
    - '{{ keys }}'

  tasks:
    - name: PUT/upload with metadata
      aws_s3:
        object: '{{ item.name }}'
        src: '{{ item.name }}'
        mode: '{{ item.state }}'
        metadata: '{{ item.meta_data }}'
      loop: '{{ input }}'
```

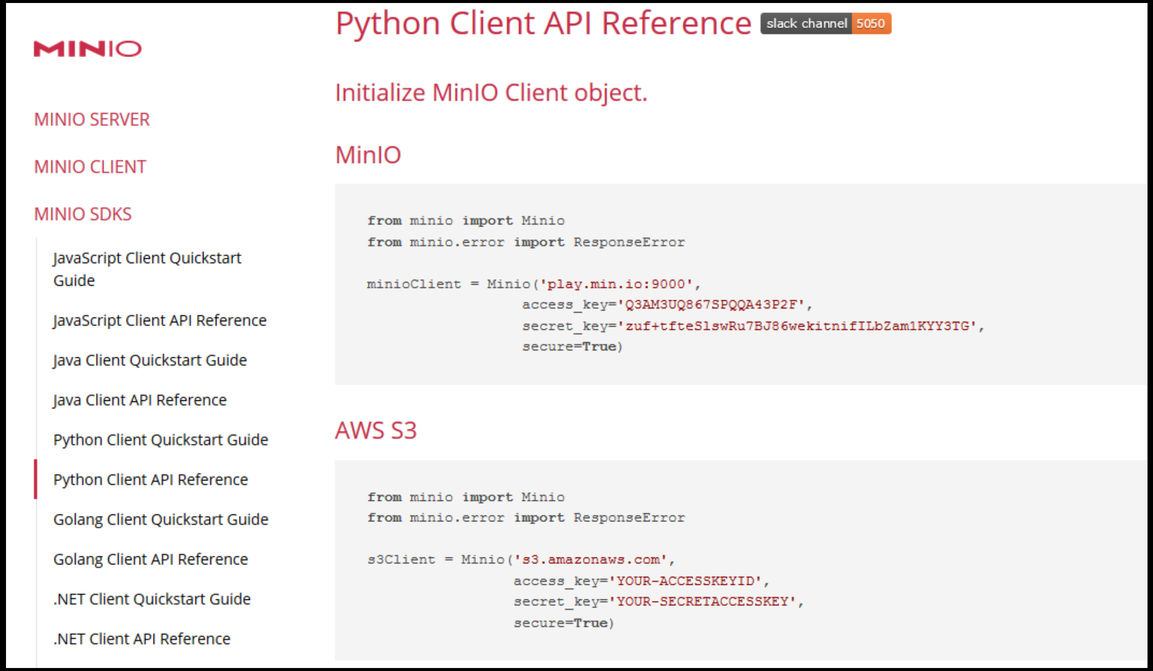
minio.io

https://min.io/product

For Reference

MinIO is best suited for high performance tasks such as analytics, machine learning and modern, high throughput application development. …

… As an object store, MinIO can store unstructured data such as photos, videos, log files, backups and container / VM images. Size of an object can range from a few KBs to a maximum of 5TB.



MINIO
MINIO SERVER
MINIO CLIENT
MINIO SDKS
JavaScript Client Quickstart Guide
JavaScript Client API Reference
Java Client Quickstart Guide
Java Client API Reference
Python Client Quickstart Guide
Python Client API Reference
Golang Client Quickstart Guide
Golang Client API Reference
.NET Client Quickstart Guide
.NET Client API Reference
slack channel 5050

Python Client API Reference

Initialize MinIO Client object.

MinIO

```
from minio import Minio
from minio.error import ResponseError

minioClient = Minio('play.min.io:9000',
                    access_key='Q3AM3UQ867SPQQA43P2F',
                    secret_key='zuf+tfteSlswRu7BJ86wekitnifILbZam1KYY3TG',
                    secure=True)
```

AWS S3

```
from minio import Minio
from minio.error import ResponseError

s3Client = Minio('s3.amazonaws.com',
                 access_key='YOUR-ACCESSKEYID',
                 secret_key='YOUR-SECRETACCESSKEY',
                 secure=True)
```

Amazon S3 or Amazon Simple Storage Service is a service offered by Amazon Web Services that provides object storage through a web service interface.

https://en.wikipedia.org/wiki/Minio

SHARE YOUR

KNOWLEDGE
EXPERIENCE

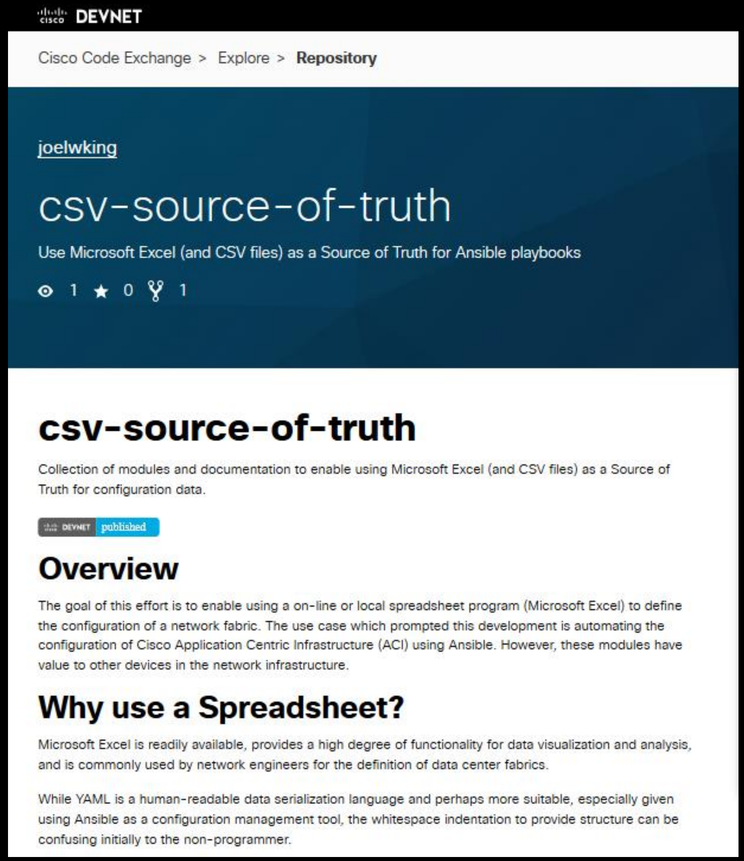
AUTOMATION: Cisco ACI

#SiliconValleyinSTL

source of truth

Configuration

- Network Engineers like spreadsheets
- Free and readily available – no training
- YAML, JSON and XML confound non-programmers



DEVNET

Cisco Code Exchange > Explore > **Repository**

joelwking

csv-source-of-truth

Use Microsoft Excel (and CSV files) as a Source of Truth for Ansible playbooks

👁 1 ★ 0 1

csv-source-of-truth

Collection of modules and documentation to enable using Microsoft Excel (and CSV files) as a Source of Truth for configuration data.

DEVNET published

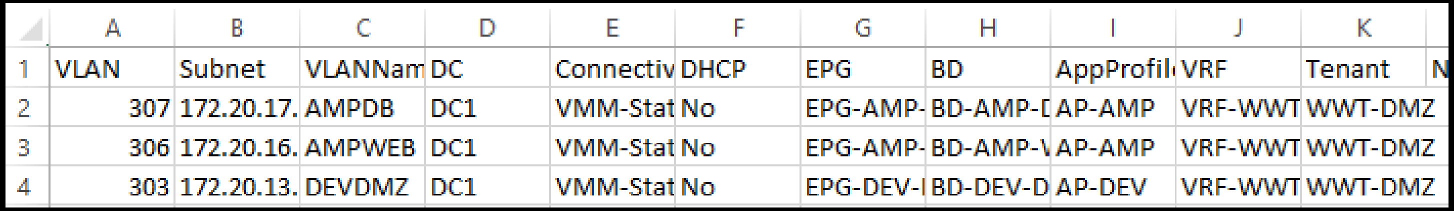
Overview

The goal of this effort is to enable using a on-line or local spreadsheet program (Microsoft Excel) to define the configuration of a network fabric. The use case which prompted this development is automating the configuration of Cisco Application Centric Infrastructure (ACI) using Ansible. However, these modules have value to other devices in the network infrastructure.

Why use a Spreadsheet?

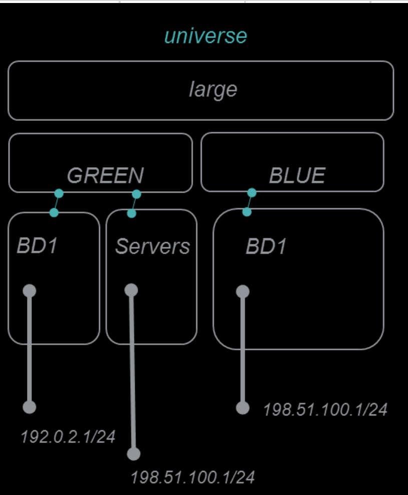
Microsoft Excel is readily available, provides a high degree of functionality for data visualization and analysis, and is commonly used by network engineers for the definition of data center fabrics.

While YAML is a human-readable data serialization language and perhaps more suitable, especially given using Ansible as a configuration management tool, the whitespace indentation to provide structure can be confusing initially to the non-programmer.



	A	B	C	D	E	F	G	H	I	J	K	
1	VLAN	Subnet	VLANNam	DC	Connectiv	DHCP	EPG	BD	AppProfil	VRF	Tenant	N
2	307	172.20.17.	AMPDB	DC1	VMM-Stat	No	EPG-AMP-	BD-AMP-I	AP-AMP	VRF-WWT	WWT-DMZ	
3	306	172.20.16.	AMPWEB	DC1	VMM-Stat	No	EPG-AMP-	BD-AMP-\	AP-AMP	VRF-WWT	WWT-DMZ	
4	303	172.20.13.	DEVDMZ	DC1	VMM-Stat	No	EPG-DEV-I	BD-DEV-D	AP-DEV	VRF-WWT	WWT-DMZ	

tabular structure to hierarchical



data optimization

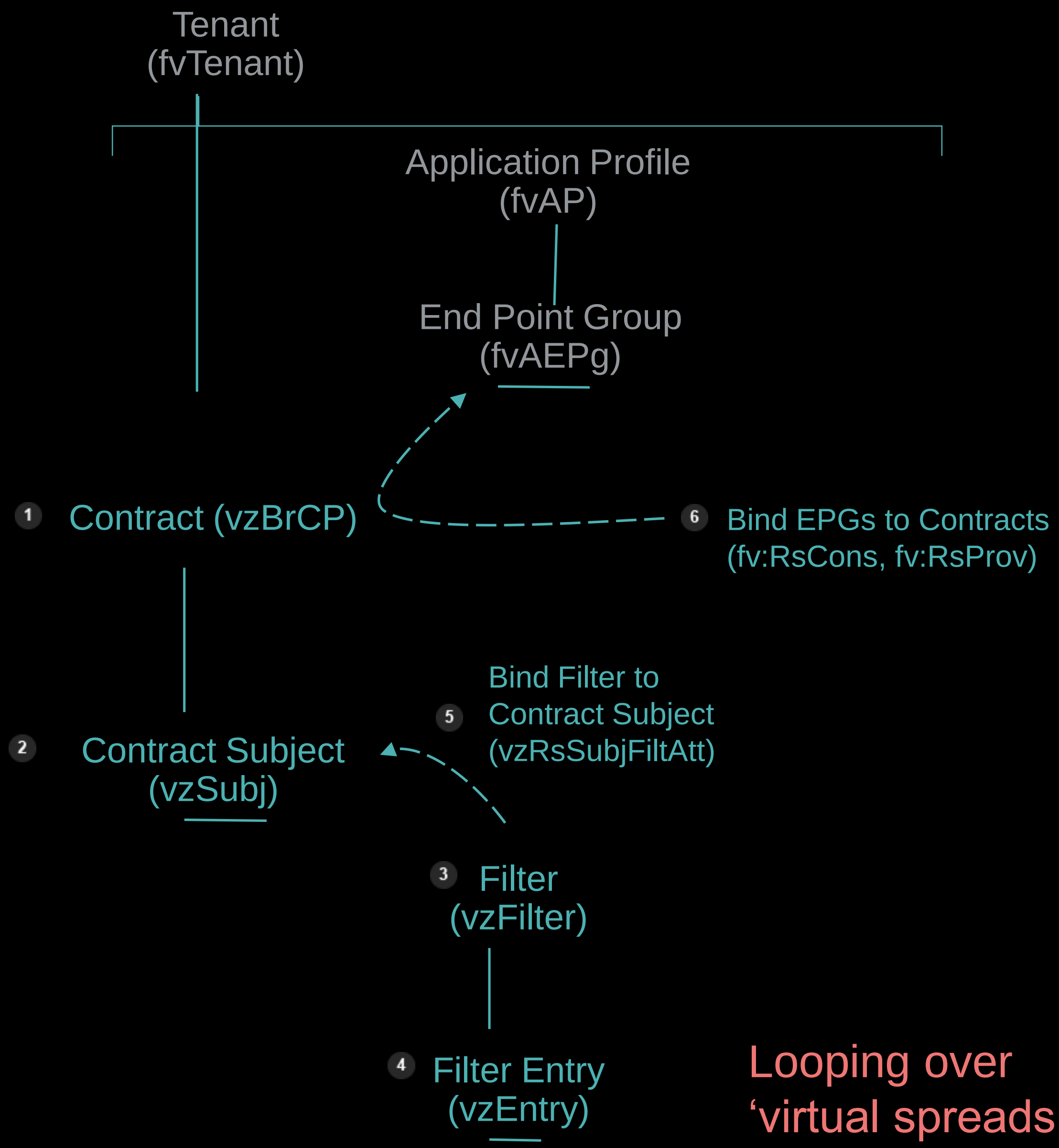
```
./csv_to_mongo.yml -e filename="/it-automation-aci/TEST_DATA/aci_constructs_policies_3.csv" --skip-tags "mongo"

  tasks:
    - name: Get facts from CSV file
      csv_to_facts:
          src: '{{ filename }}'
          table: spreadsheet
          vsheets:
            - VRFs:
              - VRF
              - Tenant
            - TENANTs:
              - Tenant
            - APs:
              - Tenant
              - AppProfile
    - debug:
        msg: '{{ TENANTs }}'


TASK [debug]
******************************************************************************************************************
ok: [localhost] => {}

MSG:

[{u'Tenant': u'WWT-INT'}, {u'Tenant': u'WWT-DMZ'}, {u'Tenant': u'WWT_NULL'}]
```

CSV FILE FROM
EXCEL

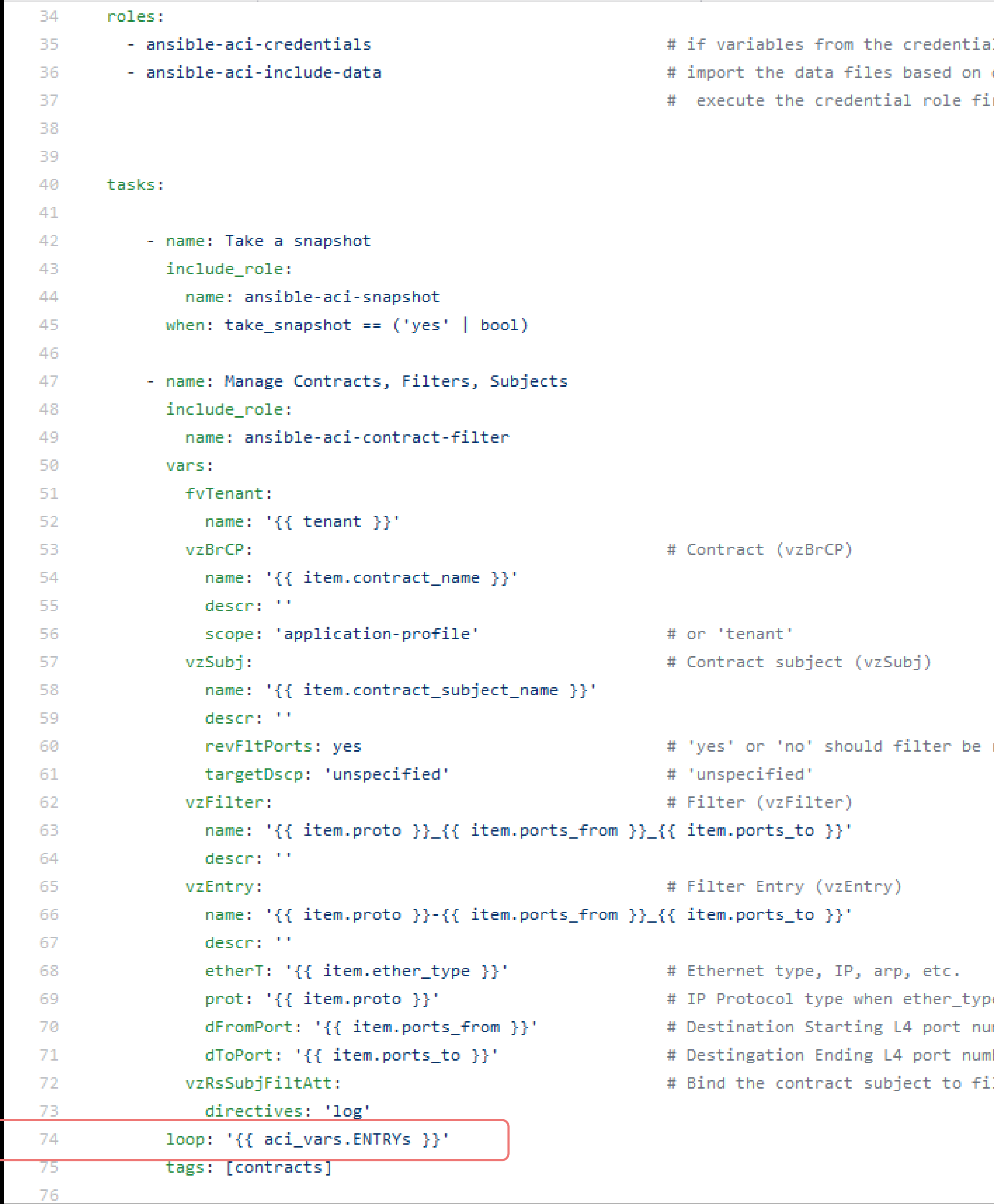
VIRTUAL SPREADSHEET

UNIQUE (SET) OF
TENANTS
PRESENT IN THE
SPREADSHEET

https://github.com/joelwking/csv-source-of-truth



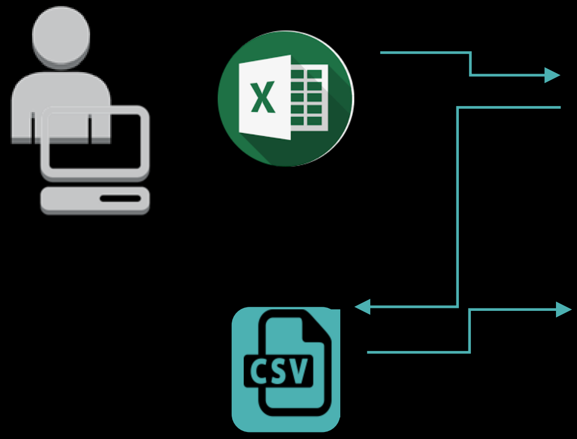
Looping over
'virtual spreadsheet



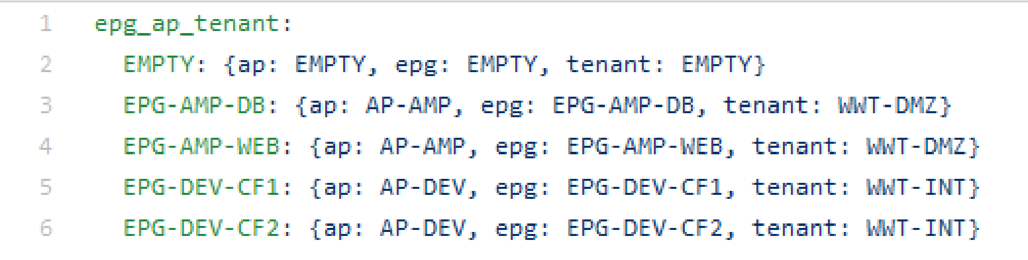
```
roles:
  - ansible-aci-credentials                              # if variables from the credentia
  - ansible-aci-include-data                             # import the data files based on
                                                         #  execute the credential role fi


tasks:

    - name: Take a snapshot
      include_role:
        name: ansible-aci-snapshot
      when: take_snapshot == ('yes' | bool)

    - name: Manage Contracts, Filters, Subjects
      include_role:
        name: ansible-aci-contract-filter
      vars:
        fvTenant:
          name: '{{ tenant }}'
        vzBrCP:                                          # Contract (vzBrCP)
          name: '{{ item.contract_name }}'
          descr: ''
          scope: 'application-profile'                   # or 'tenant'
        vzSubj:                                          # Contract subject (vzSubj)
          name: '{{ item.contract_subject_name }}'
          descr: ''
          revFltPorts: yes                               # 'yes' or 'no' should filter be
          targetDscp: 'unspecified'                      # 'unspecified'
        vzFilter:                                        # Filter (vzFilter)
          name: '{{ item.proto }}_{{ item.ports_from }}_{{ item.ports_to }}'
          descr: ''
        vzEntry:                                         # Filter Entry (vzEntry)
          name: '{{ item.proto }}-{{ item.ports_from }}_{{ item.ports_to }}'
          descr: ''
          etherT: '{{ item.ether_type }}'                # Ethernet type, IP, arp, etc.
          prot: '{{ item.proto }}'                       # IP Protocol type when ether_typ
          dFromPort: '{{ item.ports_from }}'             # Destination Starting L4 port nu
          dToPort: '{{ item.ports_to }}'                 # Destingation Ending L4 port num
        vzRsSubjFiltAtt:                                 # Bind the contract subject to fi
          directives: 'log'
      loop: '{{ aci_vars.ENTRYs }}'
      tags: [contracts]
```

© World Wide Technology

For Reference



```
  #
  #  This Excel / CSV file is managed by network engineering, it identifies the target Tenant, AP and EPG
  #
- block:
    - name: Extract the specified sheet from the Excel file
      xls_to_csv:
        src: '{{ spreadsheet }}'
        dest: '{{ playbook_dir }}/files/aci/'
        sheets: '{{ sheet_name }}'

    - name: Load the EPG, AP and Tenant mapping
      csv_to_facts:
        src: '{{ playbook_dir }}/files/aci/{{ sheet_name }}.csv'
        vsheets:
          - TENANTs:
            - ap
            - epg
            - tenant

    - name: Iterate over the spreadsheet roles, creating an associative array (dictionary) using the EPG as the key
      set_fact:
        epg_ap_tenant:  "{{ epg_ap_tenant | combine( {item.epg: item} ) }}"
      loop: "{{ TENANTs }}"
```



```
epg_ap_tenant:
  EMPTY: {ap: EMPTY, epg: EMPTY, tenant: EMPTY}
  EPG-AMP-DB: {ap: AP-AMP, epg: EPG-AMP-DB, tenant: WWT-DMZ}
  EPG-AMP-WEB: {ap: AP-AMP, epg: EPG-AMP-WEB, tenant: WWT-DMZ}
  EPG-DEV-CF1: {ap: AP-DEV, epg: EPG-DEV-CF1, tenant: WWT-INT}
  EPG-DEV-CF2: {ap: AP-DEV, epg: EPG-DEV-CF2, tenant: WWT-INT}
```

AUTOMATION: SECURITY AND COMPLIANCE

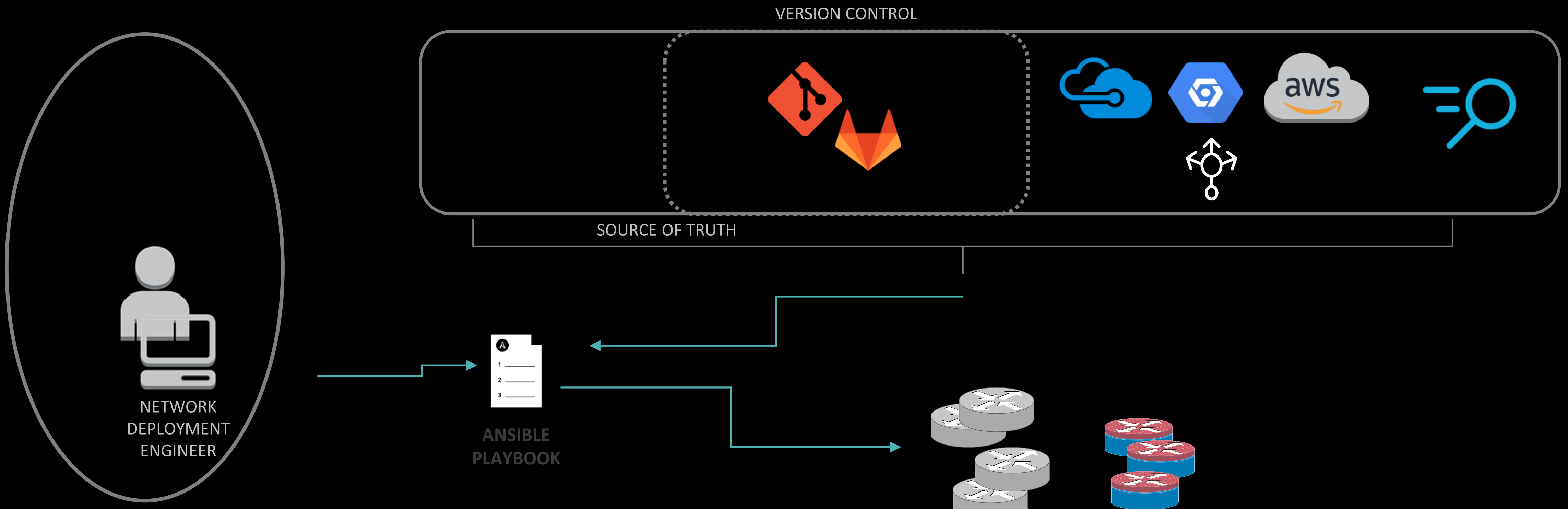
#SiliconValleyinSTL

When network configuration (policy) is generated programmatically,

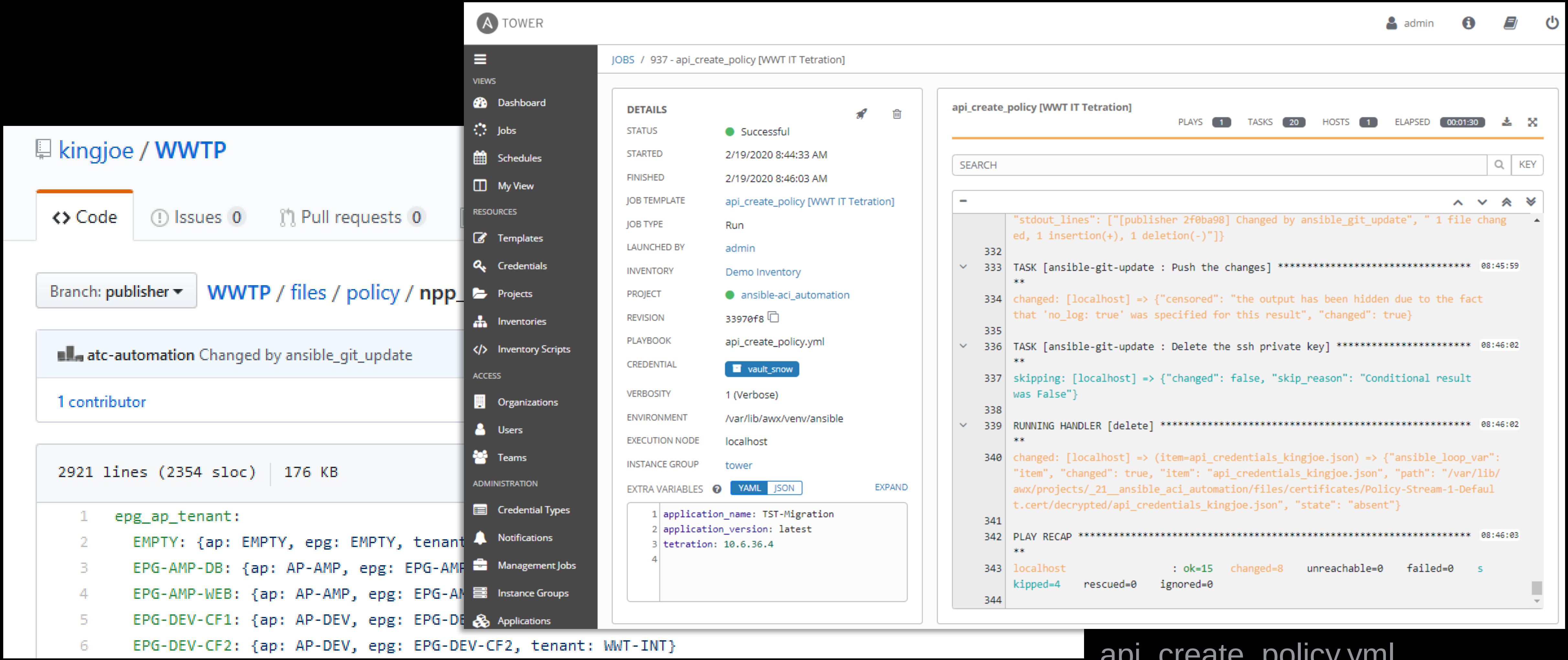
it must be applied to network devices programmatically.

Zero Trust Architecture (2nd Draft)
https://csrc.nist.gov/publications/detail/sp/800-207/draft

security automation: policy engines



demo



kingjoe / WWTP

Branch: publisher ▾ WWTP / files / policy / npp_

atc-automation Changed by ansible_git_update

1 contributor

2921 lines (2354 sloc) | 176 KB

```
epg_ap_tenant:
  EMPTY: {ap: EMPTY, epg: EMPTY, tenant
  EPG-AMP-DB: {ap: AP-AMP, epg: EPG-AMP
  EPG-AMP-WEB: {ap: AP-AMP, epg: EPG-A
  EPG-DEV-CF1: {ap: AP-DEV, epg: EPG-DE
  EPG-DEV-CF2: {ap: AP-DEV, epg: EPG-DEV-CF2, tenant: WWT-INT}
```

JOBS / 937 - api_create_policy [WWT IT Tetration]

DETAILS

STATUS ● Successful
STARTED 2/19/2020 8:44:33 AM
FINISHED 2/19/2020 8:46:03 AM
JOB TEMPLATE api_create_policy [WWT IT Tetration]
JOB TYPE Run
LAUNCHED BY admin
INVENTORY Demo Inventory
PROJECT ● ansible-aci_automation
REVISION 33970f8
PLAYBOOK api_create_policy.yml
CREDENTIAL vault_snow
VERBOSITY 1 (Verbose)
ENVIRONMENT /var/lib/awx/venv/ansible
EXECUTION NODE localhost
INSTANCE GROUP tower
EXTRA VARIABLES YAML JSON EXPAND

```
application_name: TST-Migration
application_version: latest
tetration: 10.6.36.4
```

api_create_policy [WWT IT Tetration]

PLAYS 1 TASKS 20 HOSTS 1 ELAPSED 00:01:30

```
"stdout_lines": ["[publisher 2f0ba98] Changed by ansible_git_update", " 1 file chang
ed, 1 insertion(+), 1 deletion(-)"]}

TASK [ansible-git-update : Push the changes] ********************************* 08:45:59
**
changed: [localhost] => {"censored": "the output has been hidden due to the fact
that 'no_log: true' was specified for this result", "changed": true}

TASK [ansible-git-update : Delete the ssh private key] ********************** 08:46:02
**
skipping: [localhost] => {"changed": false, "skip_reason": "Conditional result
was False"}

RUNNING HANDLER [delete] ***************************************************** 08:46:02
**
changed: [localhost] => (item=api_credentials_kingjoe.json) => {"ansible_loop_var":
"item", "changed": true, "item": "api_credentials_kingjoe.json", "path": "/var/lib/
awx/projects/_21__ansible_aci_automation/files/certificates/Policy-Stream-1-Defaul
t.cert/decrypted/api_credentials_kingjoe.json", "state": "absent"}

PLAY RECAP ******************************************************************* 08:46:03
**
localhost                  : ok=15   changed=8    unreachable=0    failed=0    s
kipped=4    rescued=0    ignored=0
```

api_create_policy.yml

key-points

- Most Network Engineers are not familiar with data serialization formats – which is why we use spreadsheets.

- Don’t clutter playbooks with conditionals validating data format

- Playbooks are not intended to be programming languages, write modules, and plugins.

- Structure your data so it drives the functionality of the playbooks

resources

Examples from the Cisco Data Center webinar
https://gitlab.com/joelwking/cisco_dc_community_of_interest

CSV source of truth
https://github.com/joelwking/csv-source-of-truth

Foray into Ansible Content Collections
https://www.slideshare.net/joelwking/foray-into-ansible-content-collections

YAML Magic
https://www.slideshare.net/roidelapluie/yaml-magic

The 100% Open Source, Enterprise-Grade, Amazon S3 Compatible Object Storage
https://min.io/
https://docs.minio.io/docs/minio-quickstart-guide.html

World Wide Technology