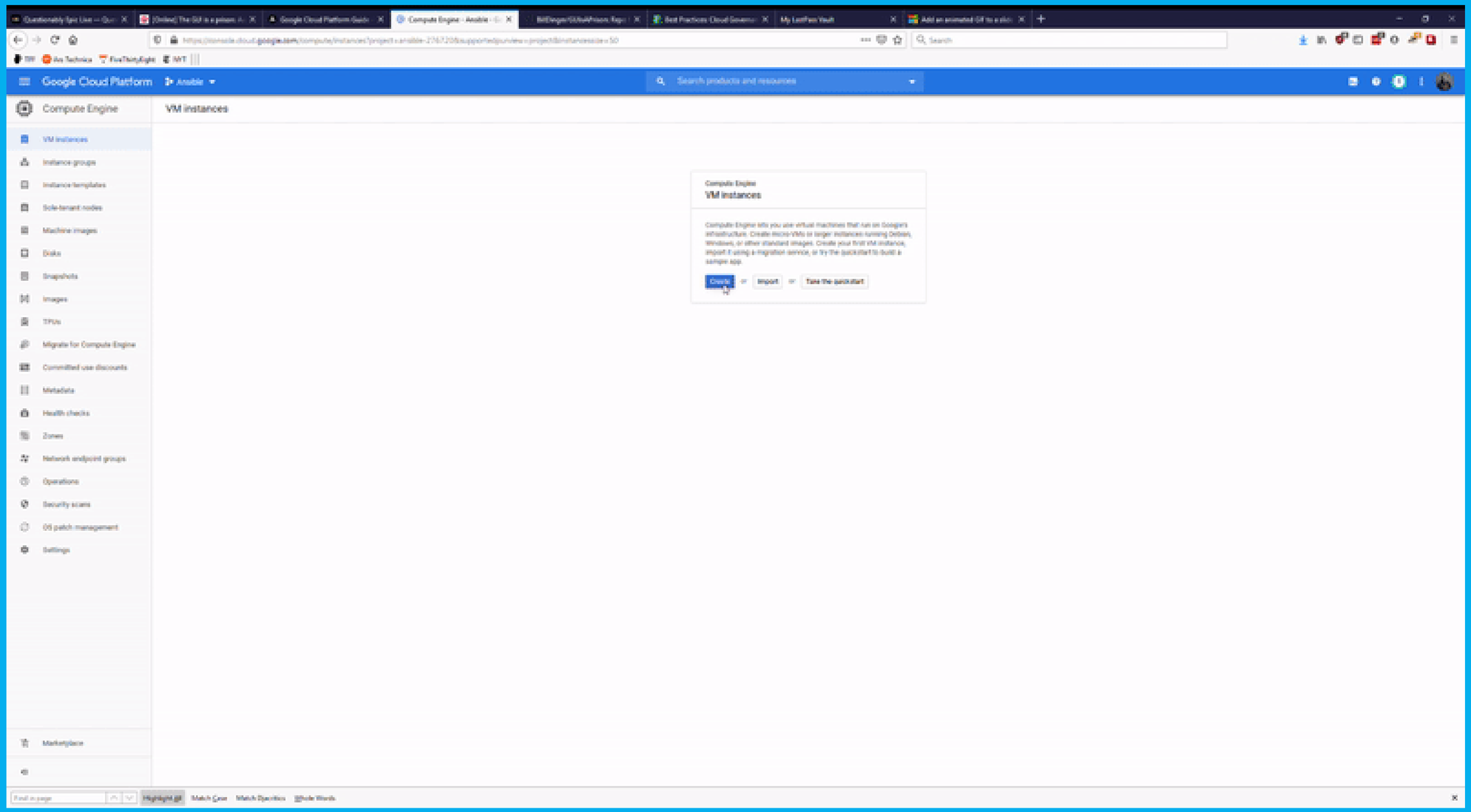
# THE GUI IS A PRISON: AUTOMATE YOUR INFRASTRUCTURE WITH ANSIBLE


BILL DINGER
Managing Directory, Technology| @adazlian
wdinger@gmail.com
https://github.com/billdinger/GUIisAPrison

# IN THE BEGINNING



# PROBLEMS WITH A GUI

- **Repeatability –** GUI is a single person clicking and entering information. Hard to guarantee no mistakes made.
- **Review–** only way for peer review is to manually watch as you create infrastructure.
- **Audit**- verifying infrastructure is configured correctly must be done manually.
- **DevOps –** Hard to integrate automated deployments to cloud infrastructure without way of deploying infrastructure itself as code.
- **Inconsistency –** different environments/machines might be configured in subtle, hard to detect ways.

# WHAT WE WANT

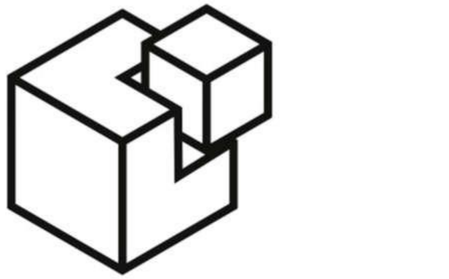
1. Human readable infrastructure as code
2. Automation of infrastructure creation
3. Auditable
4. Workflow orchestration
5. Multiplatform
6. Desired State
7. Secrets management

# OPTIONS?

Deployment Manager


CHEF™


ANSIBLE


# THINGS THEY ALL DO

- Human readable code (YML, JSON, Custom DSLs)
- Integrate into the major products / major clouds
- Allow workflow orchestration
- Provide paid & free tiers.
- Allow verification of infrastructure (Desired State)

# SO WHY ANSIBLE?

1. Core product is completely open source & free.

2. Agentless.

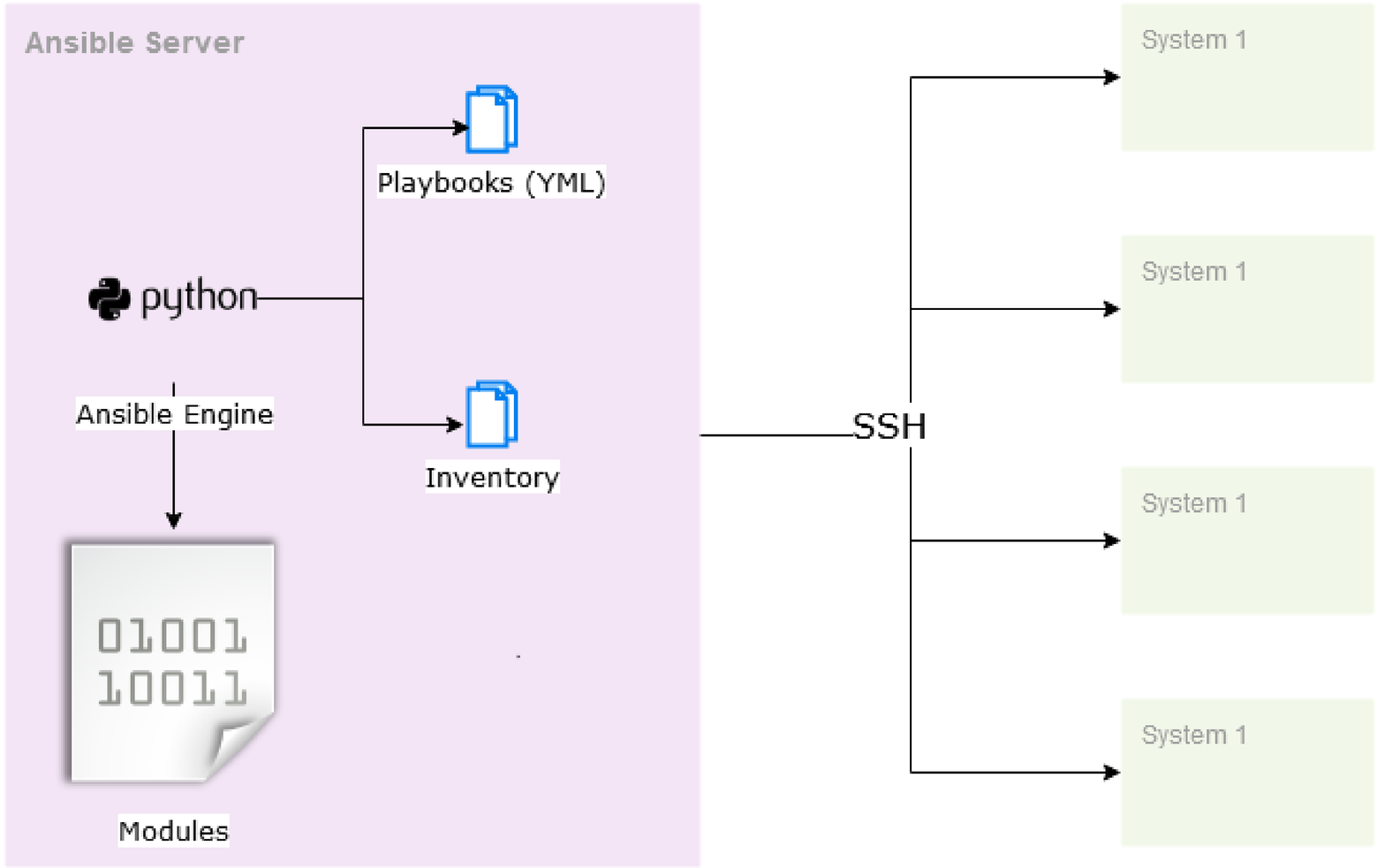
3. Supports every cloud, every major product.

4. Scores highly in Forrester, Gartner reports.

5. Python based under the hood.

6. Cloud resources & other infrastructure.

# ANSIBLE ARCHITECTURE



# ANSIBLE ARCHITECTURE – CONTROL NODE

A *nix machine running Python and has SSH installed.

Example Dockerfile:

```
FROM CENTOS:8

# INSTALL ANSIBLE + GCP STUFFS.
RUN YUM INSTALL -Y HTTPS://DL.FEDORAPROJECT.ORG/PUB/EPEL/EPEL-RELEASE-LATEST-8.NOARCH.RPM && \
YUM -Y UPDATE && \
YUM INSTALL -Y PYTHON3-PIP.NOARCH && \
PIP3 INSTALL --NO-CACHE-DIR --NO-COMPILE ANSIBLE && \
PIP3 INSTALL --NO-CACHE-DIR --NO-COMPILE REQUESTS GOOGLE-AUTH && \
PIP3 INSTALL --NO-CACHE-DIR --NO-COMPILE ANSIBLE-LINT && \
YUM INSTALL -Y NANO && \
YUM INSTALL -Y OPENSSH-CLIENTS && \
RM -RF /ROOT/.CACHE && \
FIND /USR/LIB/ -NAME '__PYCACHE__' -PRINT0 | XARGS -0 -N1 RM -RF && \
FIND /USR/LIB/ -NAME '*.PYC' -PRINT0 | XARGS -0 -N1 RM -RF

# COPY OVER SSH KEYS
RUN MKDIR -P /ROOT/.SSH && \
    CHMOD 0700 /ROOT/.SSH
ADD ./GCP.PRIVATE /ROOT/.SSH/ID_RSA
ADD ./GCP.PUB /ROOT/.SSH/ID_RSA.PUB
RUN  CHMOD 600 /ROOT/.SSH/ID_RSA && \
    CHMOD 600 /ROOT/.SSH/ID_RSA.PUB
RUN EVAL "$(SSH-AGENT -S)" && SSH-ADD

# MOUNT CURRENT WORKDIR AND START.
VOLUME ["/TMP/PLAYBOOK"]
WORKDIR  /TMP/PLAYBOOK
CMD ["BASH"]
```

# ANSIBLE ARCHITECTURE – MANAGED NODE

- Any computer of infrastructure that Ansible can command.
- SSH
- Includes windows , Linux, cloud , appliances, SaaS, etc

# ANSIBLE ARCHITECTURE – INVENTORY FILE

Example Inventory:

playbooks > ! inventory-d.yml

```yaml
---
all:
  hosts:
    ohcimgsxapp01d:
      ansible_host: 10.31.13.12
    ohcimgsxapp02d:
      ansible_host: 10.31.13.18
    ohcimgsxweb01d:
      ansible_host: 10.31.73.10
    ohcimgsxweb02d:
      ansible_host: 10.31.73.19
    OHCIMGSXMQ01D:
      ansible_host: 10.31.13.31
  children:
    app:
      hosts:
        ohcimgsxapp01d:
          ansible_host: 10.31.13.12
        ohcimgsxapp02d:
          ansible_host: 10.31.13.18
    bnl:
      hosts:
        ohcimgsxweb01d:
          ansible_host: 10.31.73.10
```

# ANSIBLE ARCHITECTURE – PLAYBOOKS

Example Playbooks:

```yaml
---
- name: Demo create Network
  hosts: localhost
  gather_facts: no
  vars_files:
    - /tmp/playbook/src/demo/gcp_auth.yml
    - /tmp/playbook/src/demo/gcp_zones.yml
  tasks:
  - name: Create GCP Network
    gcp_compute_network:
      name: ansible_network_object
      auto_create_subnetworks: 'true'
      project: "{{ gcp_project }}"
      auth_kind: "{{ gcp_auth_kind }}"
      service_account_file: "{{ gcp_credentials_file }}"
      state: present
    register: gcp_network
  - name: Create a GCP Route
    gcp_compute_route:
      name: ansible_route_object
      dest_range: 192.168.6.0/24
      next_hop_gateway: global/gateways/default-internet-gateway
      network: "{{ gcp_network }}"
      project: "{{ gcp_project }}"
      auth_kind: "{{ gcp_auth_kind }}"
      service_account_file: "{{ gcp_credentials_file }}"
      state: present
```

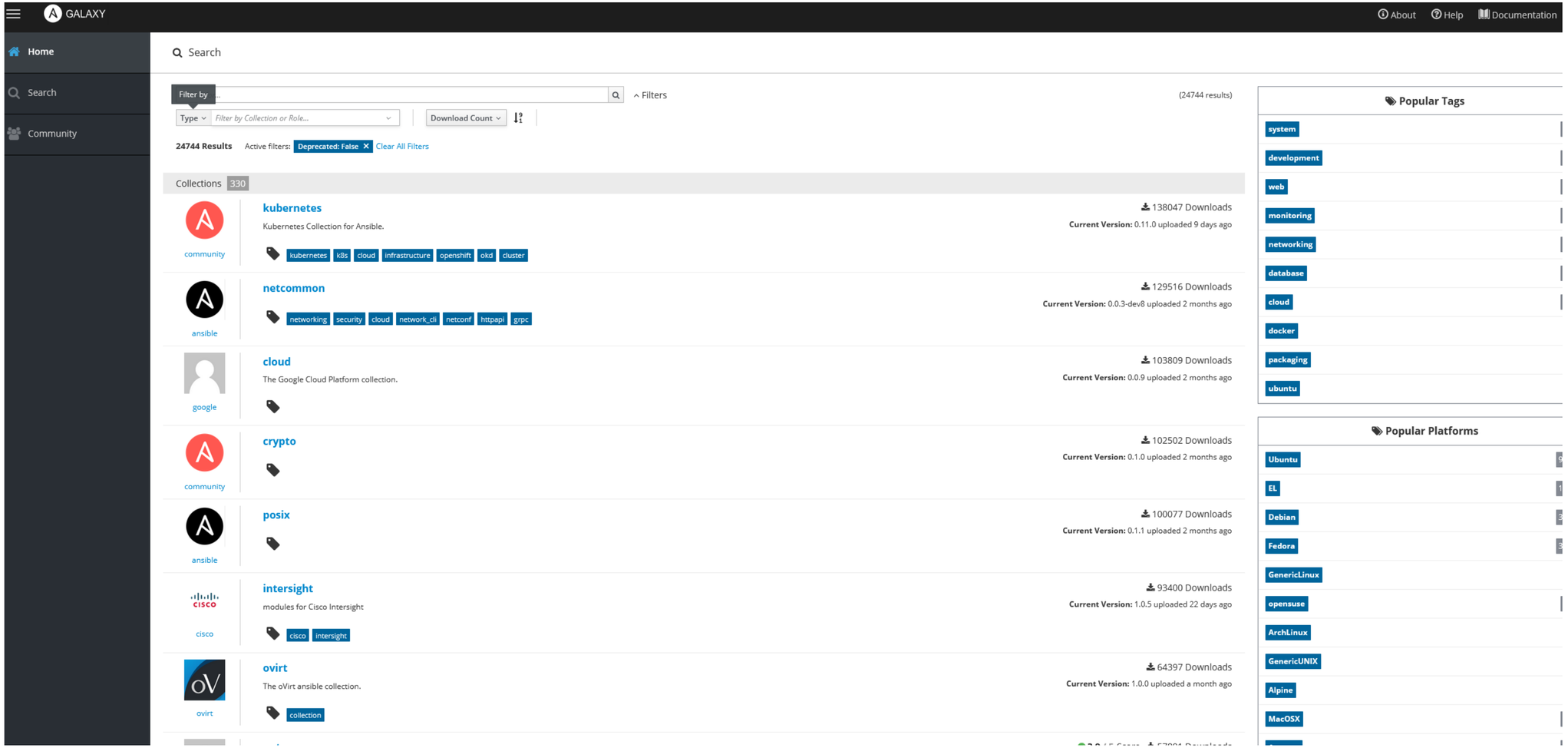
# ANSIBLE ARCHITECTURE – MODULES



## Module Index

- All modules
- Cloud modules
- Clustering modules
- Commands modules
- Crypto modules
- Database modules
- Files modules
- Identity modules
- Inventory modules
- Messaging modules
- Monitoring modules
- Net Tools modules
- Network modules
- Notification modules
- Packaging modules
- Remote Management modules
- Source Control modules
- Storage modules
- System modules
- Utilities modules
- Web Infrastructure modules
- Windows modules

# ANSIBLE ARCHITECTURE – GALAXY



# ANSIBLE ARCHITECTURE -CLI

```
ansible-playbook src/demo/gcp_tags.yml -i inventory.yml
```

```
ansible apache -a "sudo systemctl status apache2"
```

```
ansible-inventory --list -i src/demo/exampleinventory.yml
```

```
ansible-vault encrypt_string 'SuperSecretPassword'  --name 'Password'
```

# ANSIBLE ARCHITECTURE - CONFIG

```
# Example config file for ansible -- https://ansible.com/
# =====================================================

# Nearly all parameters can be overridden in ansible-playbook
# or with command line flags. Ansible will read ANSIBLE_CONFIG,
# ansible.cfg in the current working directory, .ansible.cfg in
# the home directory, or /etc/ansible/ansible.cfg, whichever it
# finds first

# For a full list of available options, run ansible-config list or see the
# documentation: https://docs.ansible.com/ansible/latest/reference_appendices/config.html.

[defaults]
#inventory      = /etc/ansible/hosts
#library        = ~/.ansible/plugins/modules:/usr/share/ansible/plugins/modules
#module_utils   = ~/.ansible/plugins/module_utils:/usr/share/ansible/plugins/module_utils
#remote_tmp     = ~/.ansible/tmp
#local_tmp      = ~/.ansible/tmp
#forks          = 5
#poll_interval  = 0.001
#ask_pass       = False
#transport      = smart

# Plays will gather facts by default, which contain information about
# the remote system.
#
# smart - gather by default, but don't regather if already gathered
# implicit - gather by default, turn off with gather_facts: False
# explicit - do not gather by default, must say gather_facts: True
#gathering = implicit

# This only affects the gathering done by a play's gather_facts directive,
# by default gathering retrieves all facts subsets
# all - gather all subsets
```

# D E M O

# WORKING WITH ANSIBLE

# ANATOMY OF A PLAYBOOK

```yaml
- name: Demo create instance
  hosts: localhost
  gather_facts: no
  vars_files:
    - /tmp/playbook/src/demo/gcp_auth.yml
    - /tmp/playbook/src/demo/gcp_zones.yml
  tasks:
  - name: create a disk
    gcp_compute_disk:
      name: disk-ansible
      size_gb: 20
      source_image: projects/centos-cloud/global/images/family/centos-8
      zone: "{{ zone }}"
      state: present
      project: "{{ gcp_project }}"
      auth_kind: "{{ gcp_auth_kind }}"
      service_account_file: "{{ gcp_credentials_file }}"
    register: disk
  - name: create a network
    gcp_compute_network:
          name: 'network-ansible'
          project: "{{ gcp_project }}"
          auth_kind: "{{ gcp_auth_kind }}"
          service_account_file: "{{ gcp_credentials_file }}"
          scopes:
            - https://www.googleapis.com/auth/compute
          state: present
    register: network
```

# ANATOMY OF A PLAYBOOK - VARIABLES & TAGS

```yaml
- name: Create sggp-{{ environment_prefix }} App Pool
  win_iis_webapppool:
      name: sggp-{{ environment_prefix }}
      attributes:
        enable32BitAppOnWin64: true
        managedPipelineMode: Integrated
        managedRuntimeVersion: v4.0
        startMode: AlwaysRunning
        processModel.identityType: SpecificUser
        processModel.userName: '{{ service_user }}'
        processModel.password: '{{ f_service_user }}'
        processModel.loadUserProfile: false
        processModel.idleTimeout: 0 # Different than what is currently out there for better perf.
        recycling.periodicRestart.schedule: "03:30:00"
        recycling.periodicRestart.time: 0
      state: present
  tags:
    - sggp
```

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

# ANATOMY OF A PLAYBOOK - VARIABLES CONTINUED

```
regex: "{{ 'ansible is awesome' | regex_search('(ansible)') }}"
ternary: "{{ (name == "Bill") | ternary('yay','boo') }}""
capital: "{{ bill | capitalize }}"
```

## Jenga2 Based:

https://docs.ansible.com/ansible/latest/user_guide/playbooks_filters.html#playbooks-filters

# ANATOMY OF A PLAYBOOK - LOOPS

```yaml
tasks:
- name: create a managed zone
  gcp_dns_managed_zone:
    name: "{{ item.name }}"
    dns_name: "{{ item.dns }}"
    description: Ansible created
    project: "{{ gcp_project }}"
    auth_kind: "{{ gcp_auth_kind }}"
    service_account_file: "{{ gcp_credentials_file }}"
    state: present
  loop:
    - { name: 'prod', dns: 'ansible.demo.com.' }
    - { name: 'stage', dns: 'stage.ansible.demo.com.' }
```

# ANATOMY OF A PLAYBOOK – RETURN VALUES

```yaml
- name: create a topic
  gcp_pubsub_topic:
    name: ansible-topic1
    project: "{{ gcp_project }}"
    auth_kind: "{{ gcp_auth_kind }}"
    service_account_file: "{{ gcp_credentials_file }}"
    state: present
  register: ansible_pubsub_output
- debug:
    var: ansible_pubsub_output
```

```
TASK [debug] ********************************************************************************
ok: [localhost] => {
    "ansible_pubsub_output": {
        "changed": true,
        "failed": false,
        "name": "projects/ansible-276720/topics/ansible-topic1"
    }
}
```

# ANATOMY OF A PLAYBOOK – HANDLERS

```
handlers:
  - name: Add tags to instance
    gce_tag:
      instance_name: "{{ instance.name}}"
      tags: ansible-tags
      zone: "{{ zone }}"
      state: present
```

# ANATOMY OF A PLAYBOOK – CONDITIONALS

```
- shell: echo "only targetting docker"
  when: ansible_facts['virtualization_type'] == "docker"
```

```
- debug: var=ansible_facts
```

# ANATOMY OF A PLAYBOOK – STATE

```
state: present
```

```
state: absent
```

# ANATOMY OF A PLAYBOOK – STATE CONTINUED

```
- name: GCP State changed?
  shell: echo '***CHANGED***'
  register: gcpStateResult
  changed_when: "'***CHANGED***'in gcpStateResult.stdout"
```

```
- name: GCP File
  shell: echo 'some stuff here' >> /tmp/state.txt
  args:
    creates: /tmp/state.txt
```

# ANATOMY OF A PLAYBOOK – BECOME

```
become: yes
become_method: runas
become_user: "{{ service_user }}"
```

# ANATOMY OF A PLAYBOOK – ROLES

```yaml
- name: Apply App plays
  hosts: web
  roles:
    - common
    - web
```

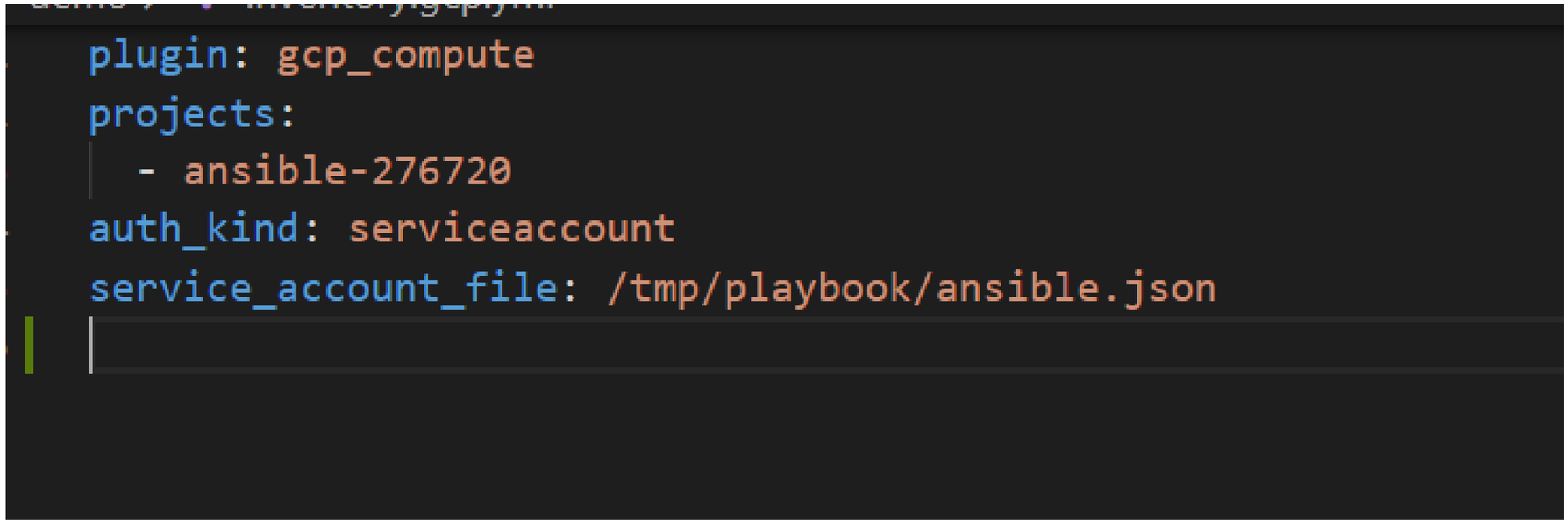
```yaml
all:
  hosts:
    WEBSERVERA:
    WEBSERVERB:
    APPSERVERA:
    APPSERVERB:
  children:
    web:
      hosts:
        WEBSERVERA:
        WEBSERVERB:
    app:
      hosts:
        APPSERVERA:
        APPSERVERB:
```

```
[root@7ac6050c9820 web]# ls -R
.:
defaults  files  handlers  meta  tasks  templates  vars

./defaults:

./files:
README.MD

./handlers:
main.yml

./meta:
main.yml

./tasks:
main.yml

./templates:
readme.md

./vars:
main.yml
```

# DEMO: GCP WALKTHROUGH

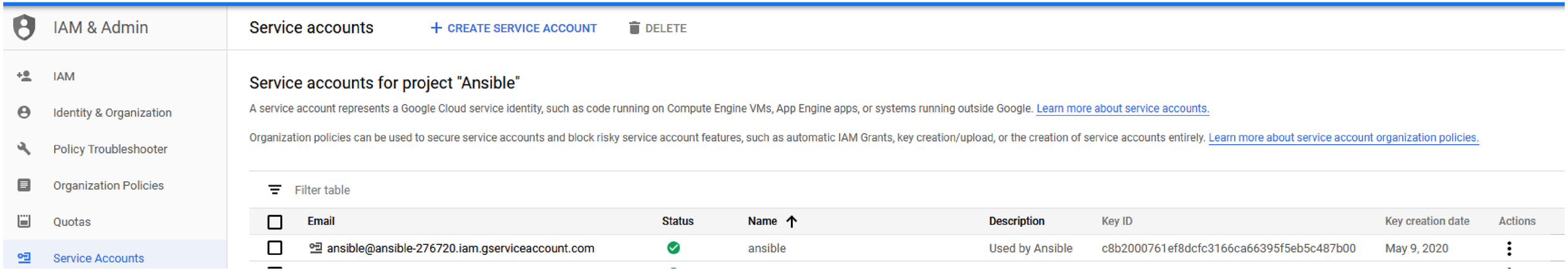
# ANSIBLE + GCP

Supports Dynamic Inventory (discovery of existing cloud assets)

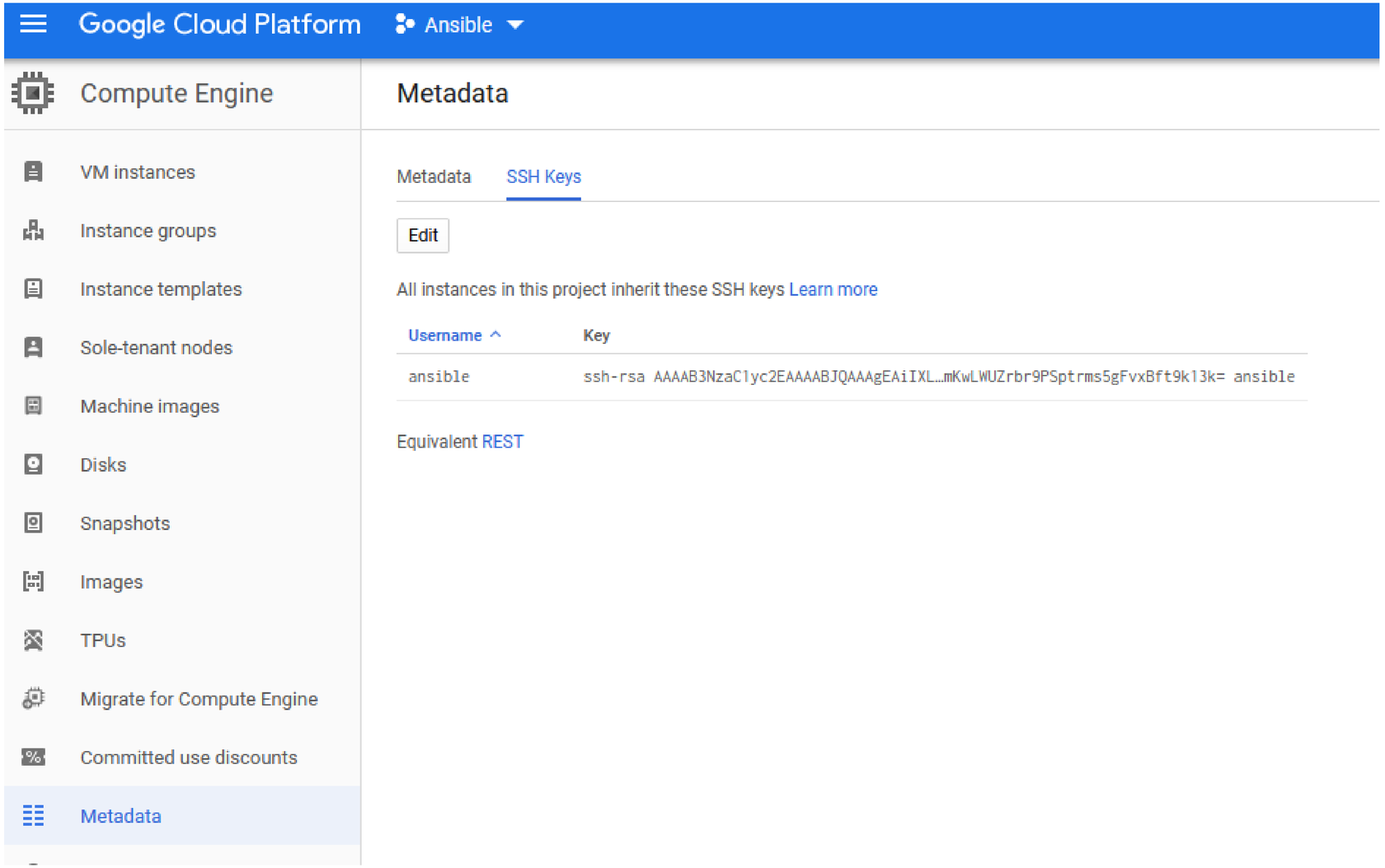


# ANSIBLE + GCP

```
gcp_project: ansible-276720
gcp_credentials_file: /tmp/playbook/ansible.json
gcp_auth_kind: serviceaccount
```



# ANSIBLE + SSH


Google Cloud Platform
Ansible
Compute Engine
Metadata
VM instances
Instance groups
Instance templates
Sole-tenant nodes
Machine images
Disks
Snapshots
Images
TPUs
Migrate for Compute Engine
Committed use discounts
Metadata
Metadata
SSH Keys
Edit
All instances in this project inherit these SSH keys Learn more
Username
Key
ansible
ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAgEAiIXL…mKwLWUZrbr9PSptrms5gFvxBft9k13k= ansible
Equivalent REST

# DEEP DIVE

# DEEP DIVE: BEST PRACTICES

- Always refer to state / Always output state
- Keep it Simple (to start)
- Roll updates
- Keep a stage/Testing  environment
- Manage _Everything_ through Ansible
- Submit PRs, run your Ansible through a CI/CD pipeline

# DEEP DIVE: DEBUGGING

```yaml
- name: create a disk
  gcp_compute_disk:
    name: disk-ansible
    size_gb: 20
    source_image: projects/centos-cloud/global/images/family/centos-8
    zone: "{{ zone }}"
    state: present
    project: "{{ gcp_project }}"
    auth_kind: "{{ gcp_auth_kind }}"
    service_account_file: "{{ gcp_credentials_file }}"
  register: disk
  debugger: on_skipped
  when: ansible_facts['virtualization_type'] == "Docker"
```

```
TASK [create a disk] ************************
skipping: [localhost]
[localhost] TASK: create a disk (debug)>
```

# DEEP DIVE: USING ANSIBLE VAULT

```
[root@7ac6050c9820 playbook]# ansible-vault encrypt_string 'SuperSecretPassword'  --name 'GDG'
New Vault password:
Confirm New Vault password:
GDG: !vault |
          $ANSIBLE_VAULT;1.1;AES256
          37316235653636343938313166306334326637636339343932306539373564643339353436373462
          37636162373631663538613836336331663237663266366660a396464326263326232643864313530
          35366331393839353866646334666663038356231656138376631616631636438303932343831376 5
          31653662653066396 60a3632316133646439316432353064663935366434663136346435623237 30
          36366233356136613837333439306631373434366166656233336631326438636130
Encryption successful
```

```
ansible-playbook src/demo/gcp_vault.yml --ask-vault-pass
```

```
ansible-playbook src/demo/gcp_vault.yml -vault-password-file something.yml
```

# DEEP DIVE: ANSIBLELINT

```
[root@2d089debaefb playbook]# ansible-lint src/demo/gcp_instance.yml
Syntax Error while loading YAML.
  expected <block end>, but found '<block mapping start>'

The error appears to be in '/tmp/playbook/src/demo/gcp_instance.yml': line 39, c
olumn 10, but may
be elsewhere in the file depending on the exact syntax problem.

The offending line appears to be:

          - https://www.googleapis.com/auth/compute
        state: present
        ^ here

[root@2d089debaefb playbook]# ansible-lint src/demo/gcp_instance.yml
[root@2d089debaefb playbook]# ansible-lint src/demo/gcp_auth.yml
[201] Trailing whitespace
src/demo/gcp_auth.yml:4
gcp_project: ansible-276720
```

# DEEP DIVE: ANSIBLE + DEVOPS PRACTICES

Integrate into your pipelines

Automatically deploy to Dev, etc on merge

Use environment variables and inside pipelines echo results to a file that ansible-playbook can use

Use ansible-lint to verify syntax on Pull Requests

# DEEP DIVE: EXTENDING ANSIBLE

Just because  you can….

https://docs.ansible.com/ansible/latest/dev_guide/developing_modules.html

Always check community first

Python3 , Powershell, or “Native”  available.

Just write a custom script.

# THANK YOU.

BILL DINGER
Managing Director, Technology @adazlian
wdinger@gmail.com
https://github.com/BillDinger/GUIisAPrison

https://docs.ansible.com/ansible/latest/user_guide/index.html

https://docs.ansible.com/ansible/latest/scenario_guides/guide_gce.html

https://github.com/ansible/ansible-examples

https://github.com/GoogleCloudPlatform/compute-video-demo-ansible