The bridge to possible

5 Tools You Should Learn to Level Up Your Network Automation Game

Kareem Iskander | Lead Technical Advocate | @kareem_isk
Quinn Snyder | Senior Technical Advocate | @qsnyder
DEVNET-2149

Cisco Webex App

Questions?

Use Cisco Webex App to chat
with the speaker after the session

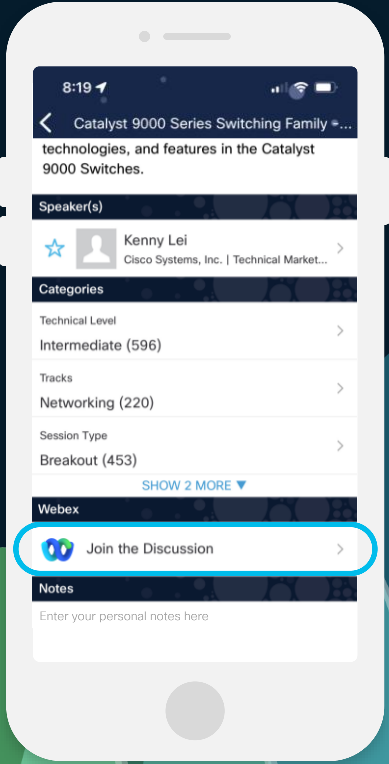
How

1. Find this session in the Cisco Live Mobile App
2. Click "Join the Discussion"
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until June 7, 2024.

cisco Live!

https://ciscolive.ciscoevents.com/ciscolivebot/#DEVNET-2149



8:19

Catalyst 9000 Series Switching Family ...

technologies, and features in the Catalyst 9000 Switches.

Speaker(s)

Kenny Lei
Cisco Systems, Inc. | Technical Market...

Categories

Technical Level
Intermediate (596)

Tracks
Networking (220)

Session Type
Breakout (453)

SHOW 2 MORE ▼

Webex

Join the Discussion

Notes

Enter your personal notes here

Agenda

- Introduction
- Git’in it right
- Its OK to be `bash`ful
- Locking the Vault
- Codifying your infra
- Containerize this!
- Conclusion

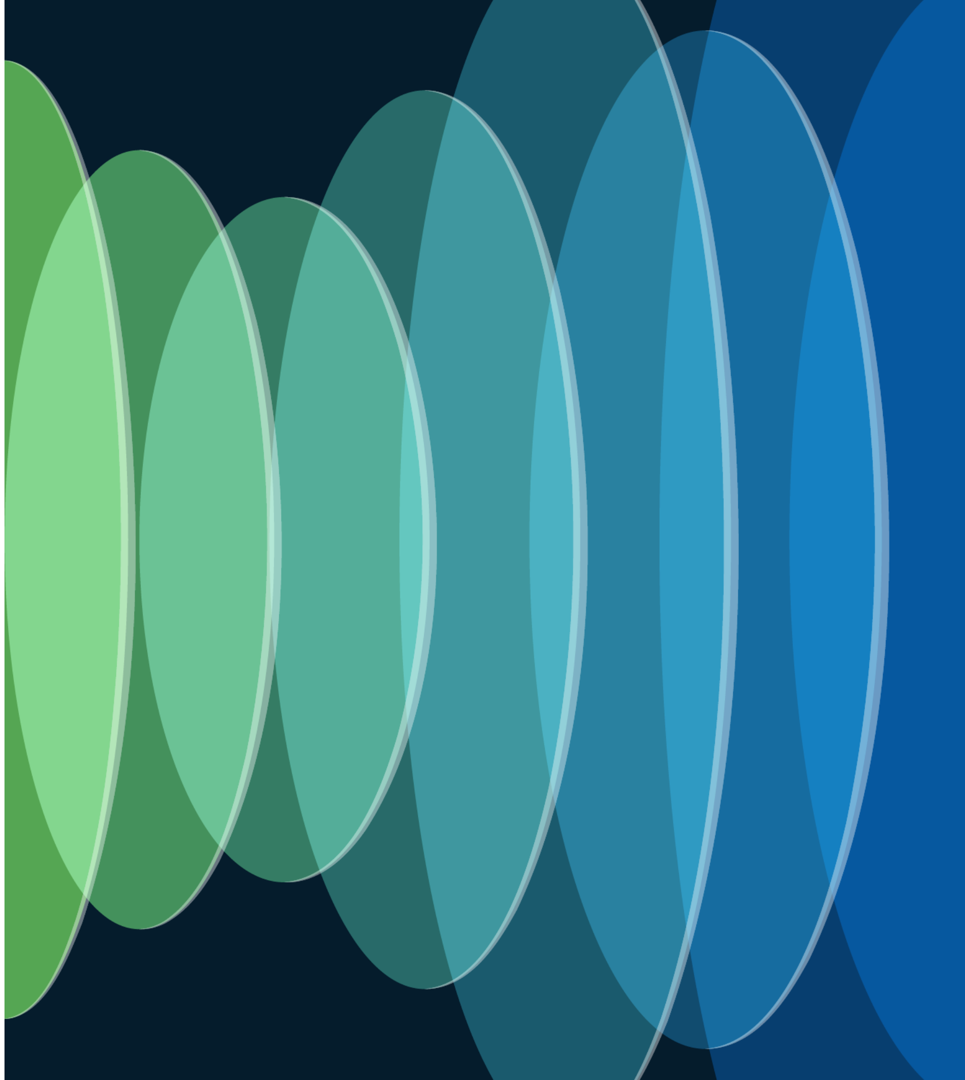
Introduction

So.. I learned the building blocks of network automation, but what happens in practice in the real world?

Let's Level Up!

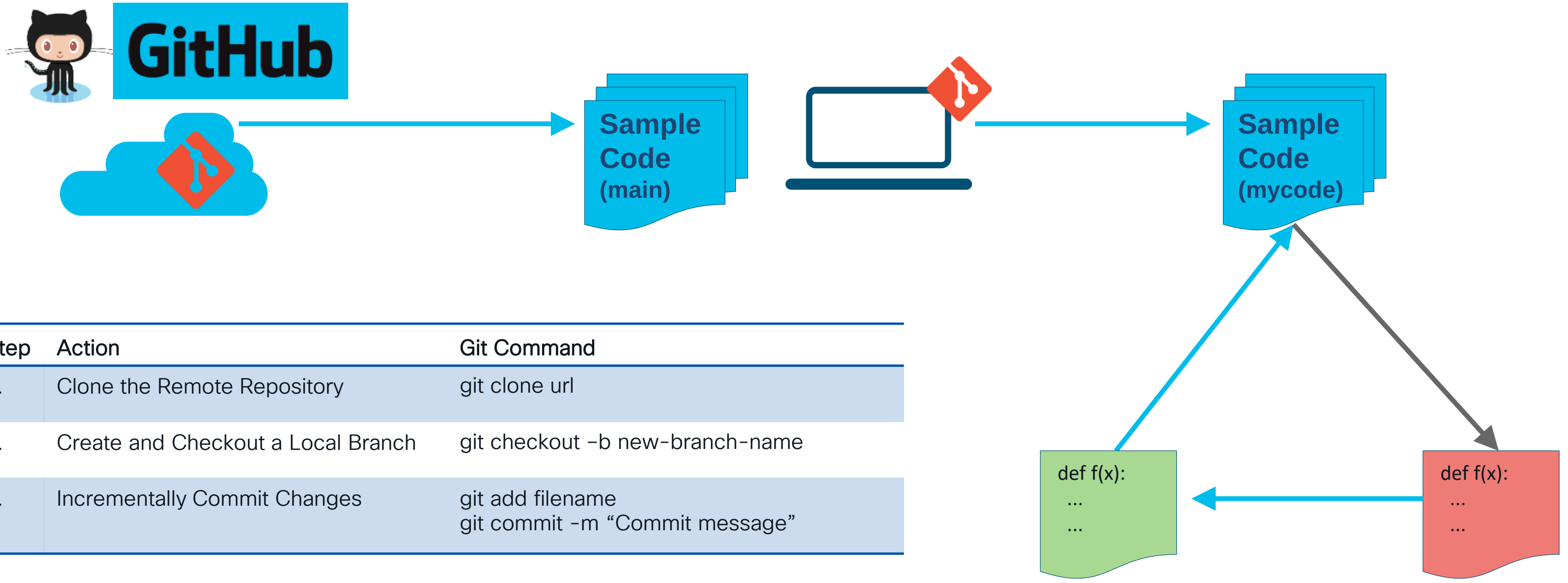
Git’in it right

What we've been talking to you about



GIT - Basics

What we've been talking to you about



Step	Action	Git Command
1.	Clone the Remote Repository	git clone url
2.	Create and Checkout a Local Branch	git checkout -b new-branch-name
3.	Incrementally Commit Changes	git add filename git commit -m "Commit message"

GIT - Basics

What we’ve been talking to you about

Pull Requests

Merge Conflict

Commits

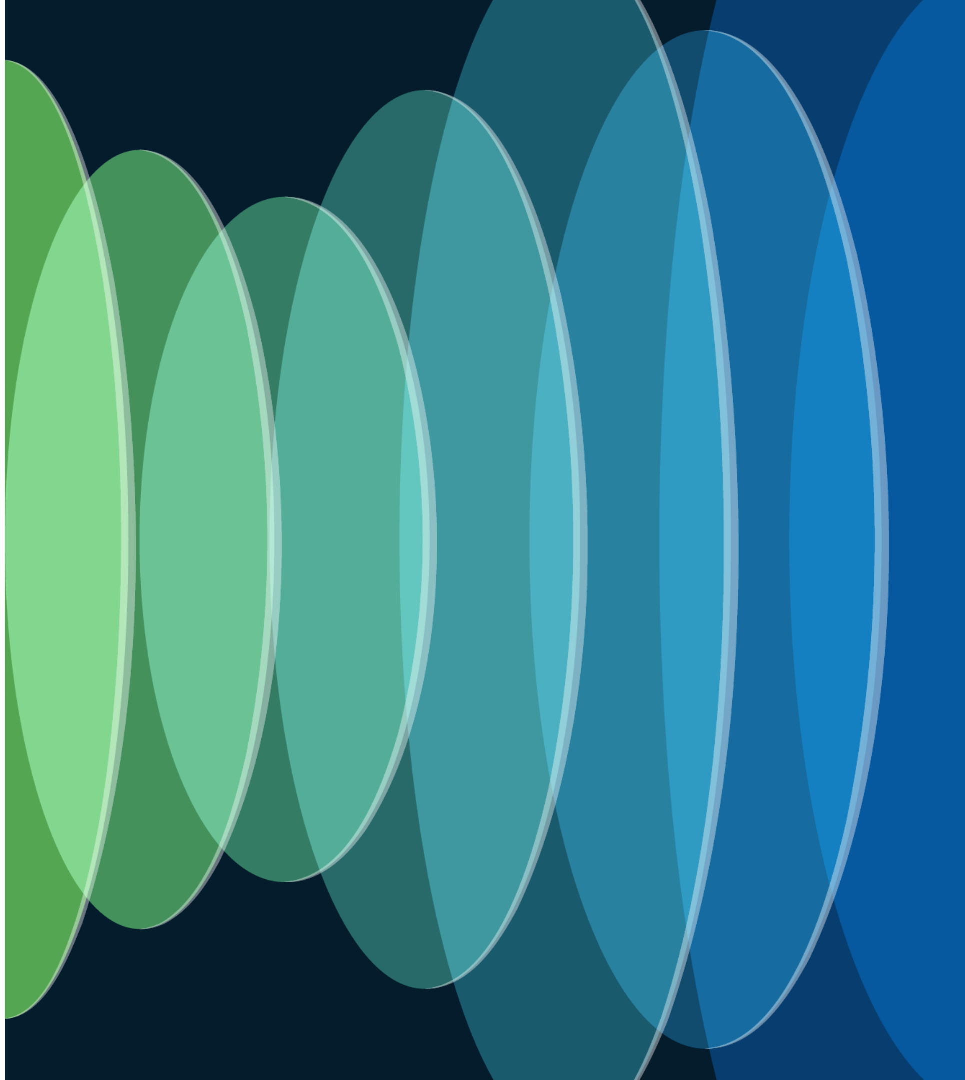
Logs

Branches

HEAD

Push

Real World



Interactive Rebase

Interactive Rebase

A Tool for Optimizing & Cleaning up Commit History

- Change a commit's message
- Delete a commit
- Reorder commits
- Combine multiple commits into one
- Edit/Split an existing commit into multiple new ones

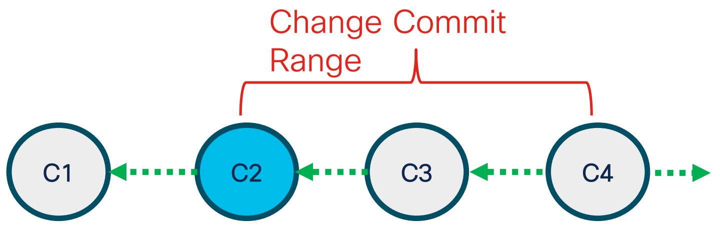
⚠ DO NOT use Interactive Rebase on commits that you've already pushed/shared on remote repo!

Interactive Rebase

Use Case

- You are working on the **Feature** branch to expand Meraki automation
- You have been making commits ever function you write
- You are ready to merge into **Main** branch
- You have realized:
    1. “Over Commit-ted” - get it?
    2. Commit messages aren’t cutting it

Interactive Rebase - Steps



1. How far back do you want to go?

2.

```
git rebase -i HEAD~2
```

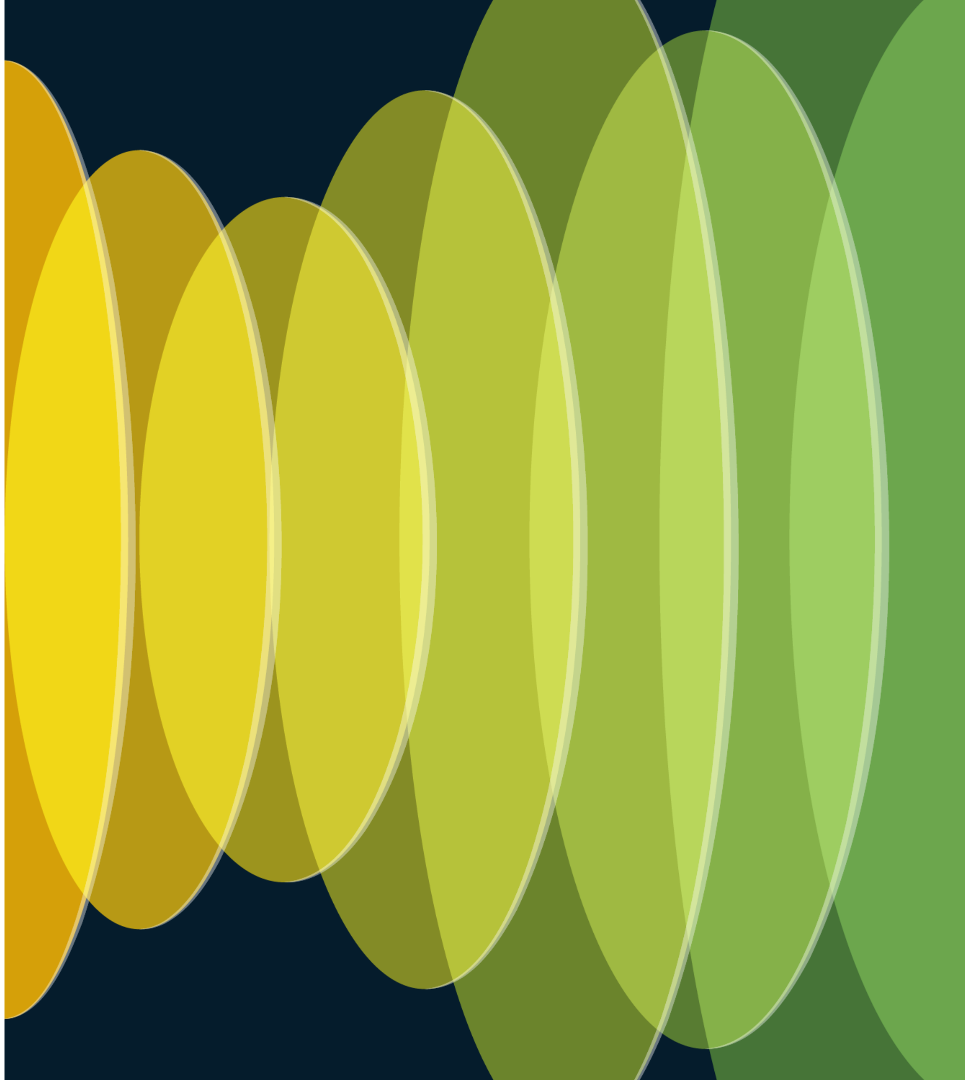
3. Determine action to apply to your commits

4. Make changes & save

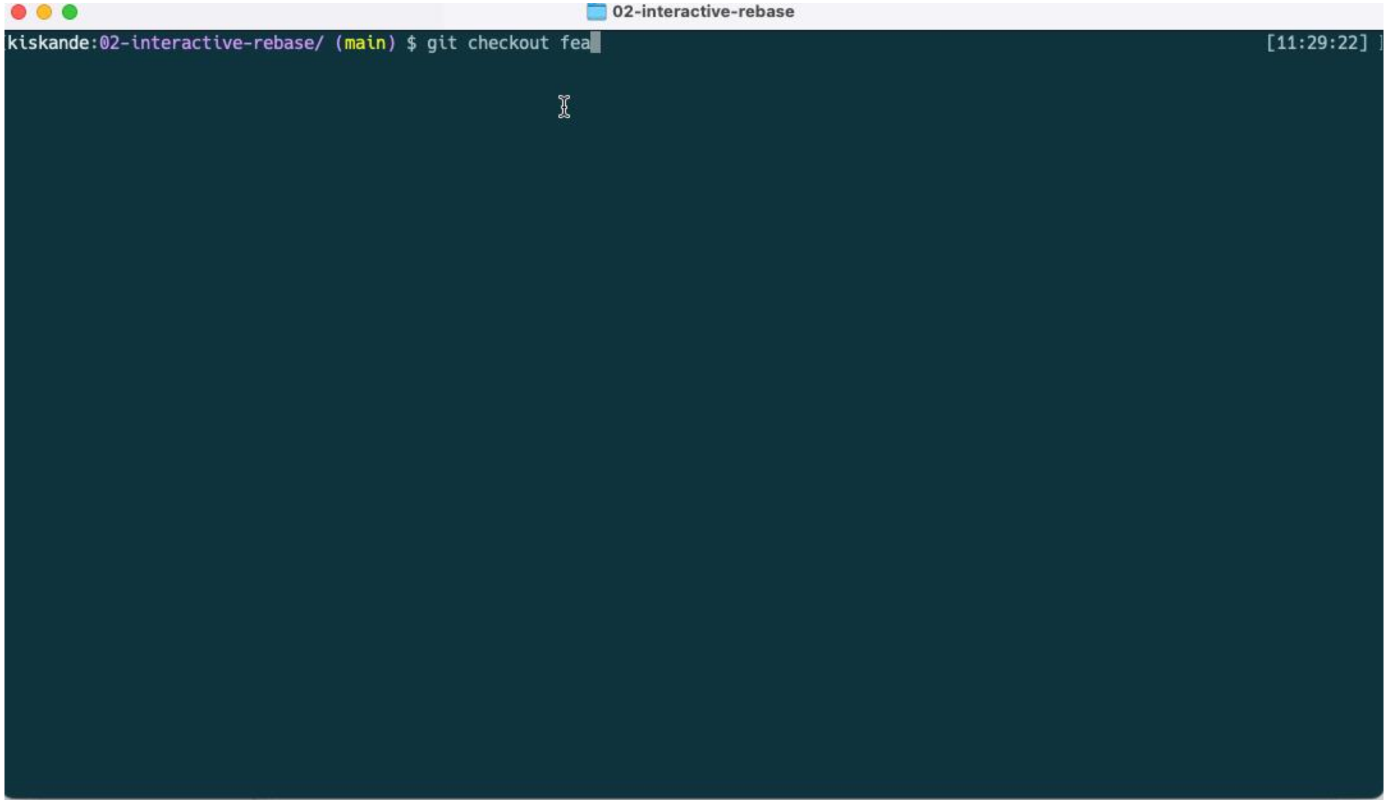
5.

```
git    log --oneline
```

Let’s try it!



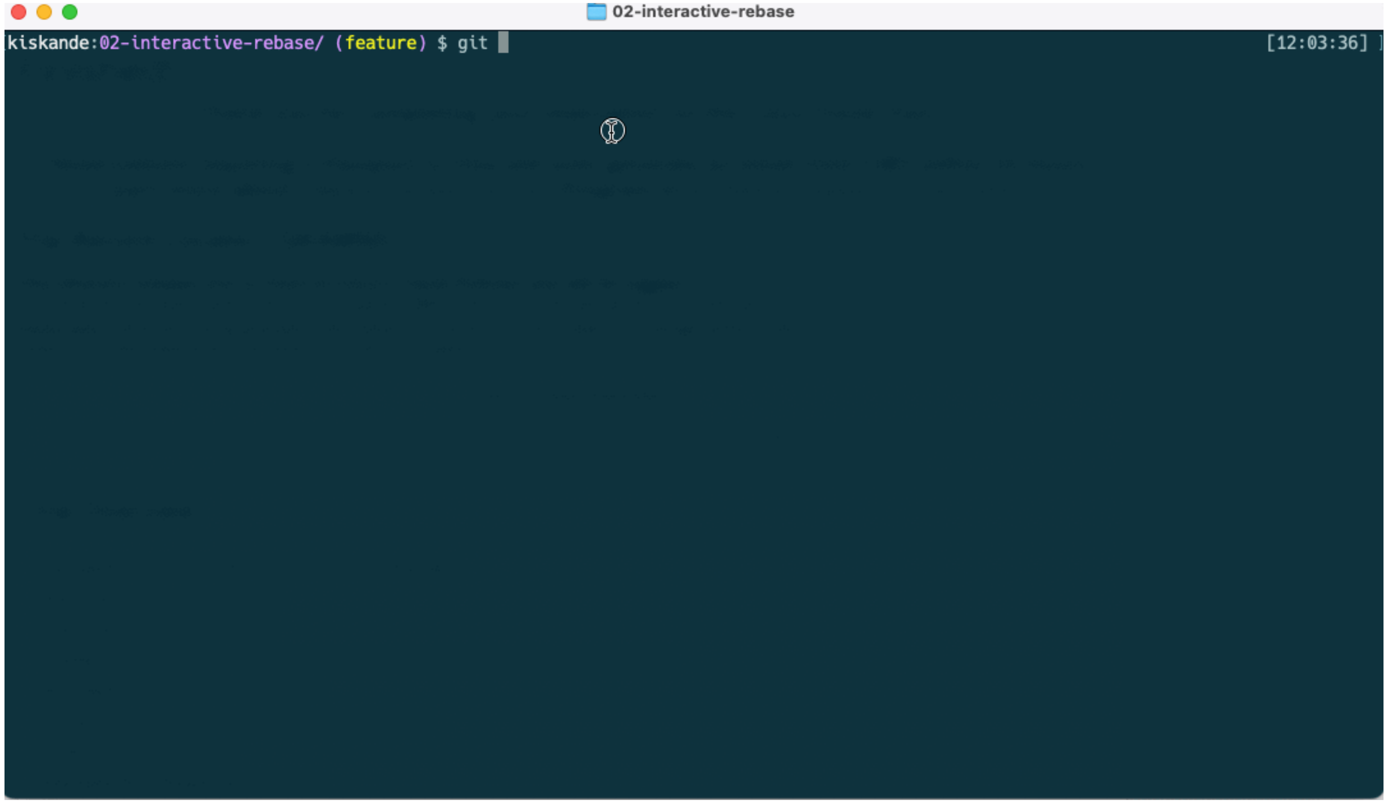
Interactive Rebase - Reword



02-interactive-rebase

```
kiskande:02-interactive-rebase/ (main) $ git checkout fea                [11:29:22]
```

Interactive Rebase - Squash


02-interactive-rebase

```
kiskande:02-interactive-rebase/ (feature) $ git                                  [12:03:36]
```

Cherry Picking

Cherry Picking 🍒

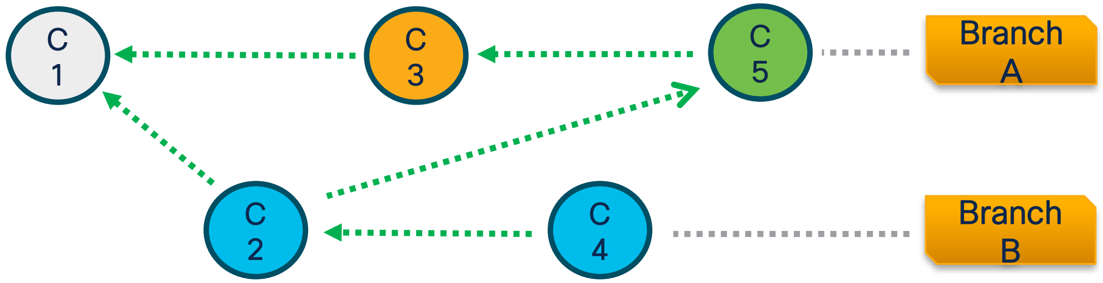
Use Case

- You are working on the **Feature** branch to expand your Meraki site automation
- You have made a commit on the **Feature** Branch
- You noticed you are working in the **Main** Branch
- The commit does not belong in **Main** .. Yet
- What to do??

Cherry Picking

Integrate Single, Specific commit

- Cherry Picking allows you to select individual commits
- Cherry Picking integrate specific commits to Branch
- Only Commit C2 from Branch B to integrate into Branch A



Cherry Picking 🍒 - Steps

1. While on **Main** Grab the commit hash

```
git    log --oneline
```

2. ranch

```
git checkout Feature
```

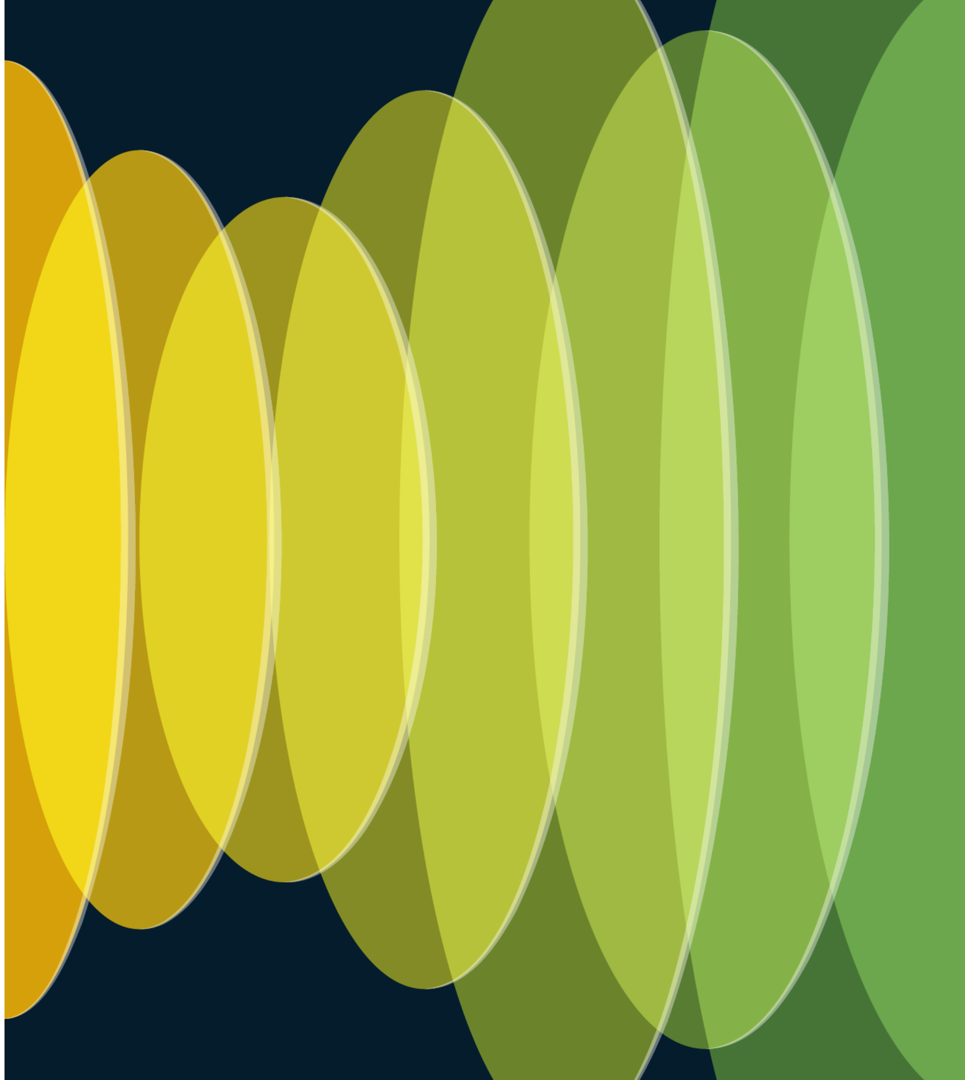
3. (

```
git cherry-pick a827df1
```

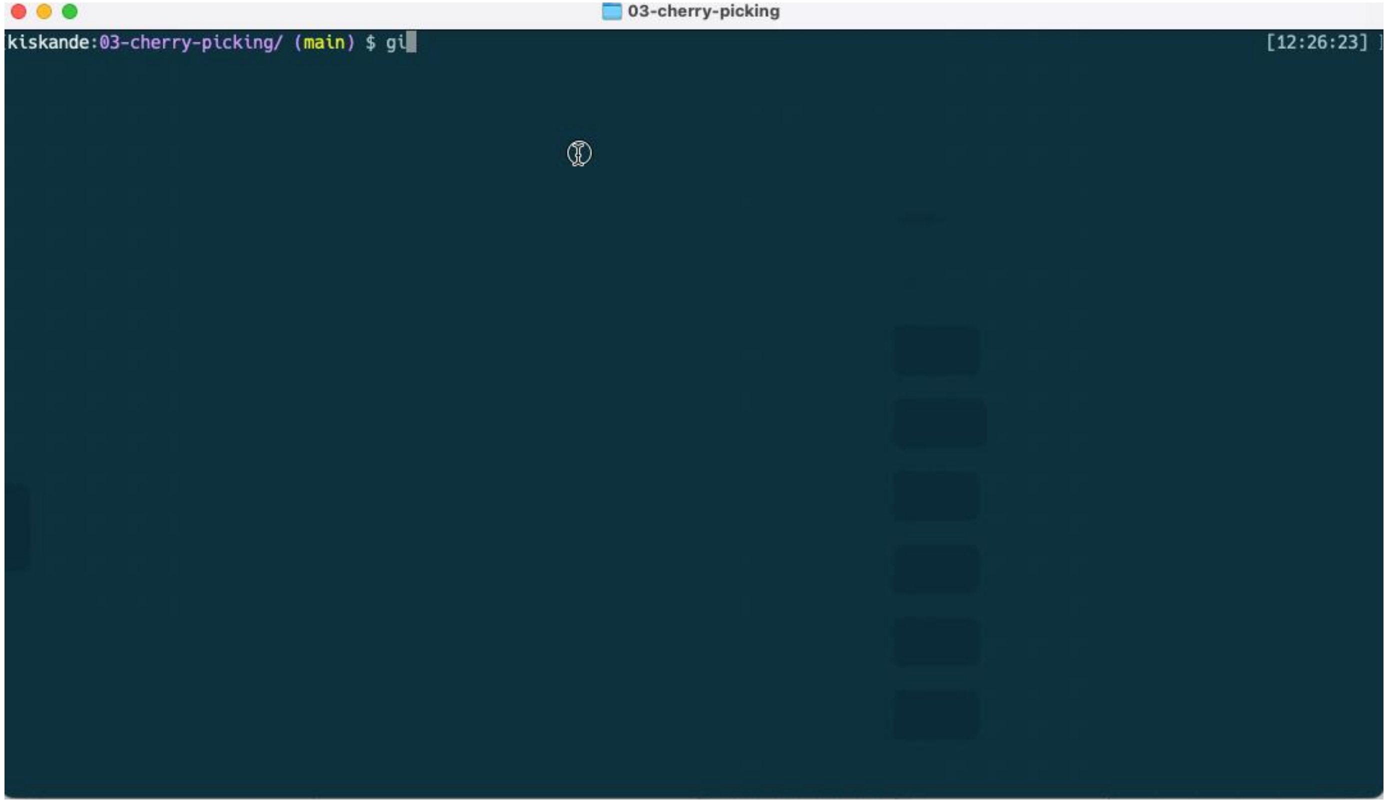
4. Clean up **Main**

```
git checkout main
git reset --hard HEAD~1
```

Let’s try it!



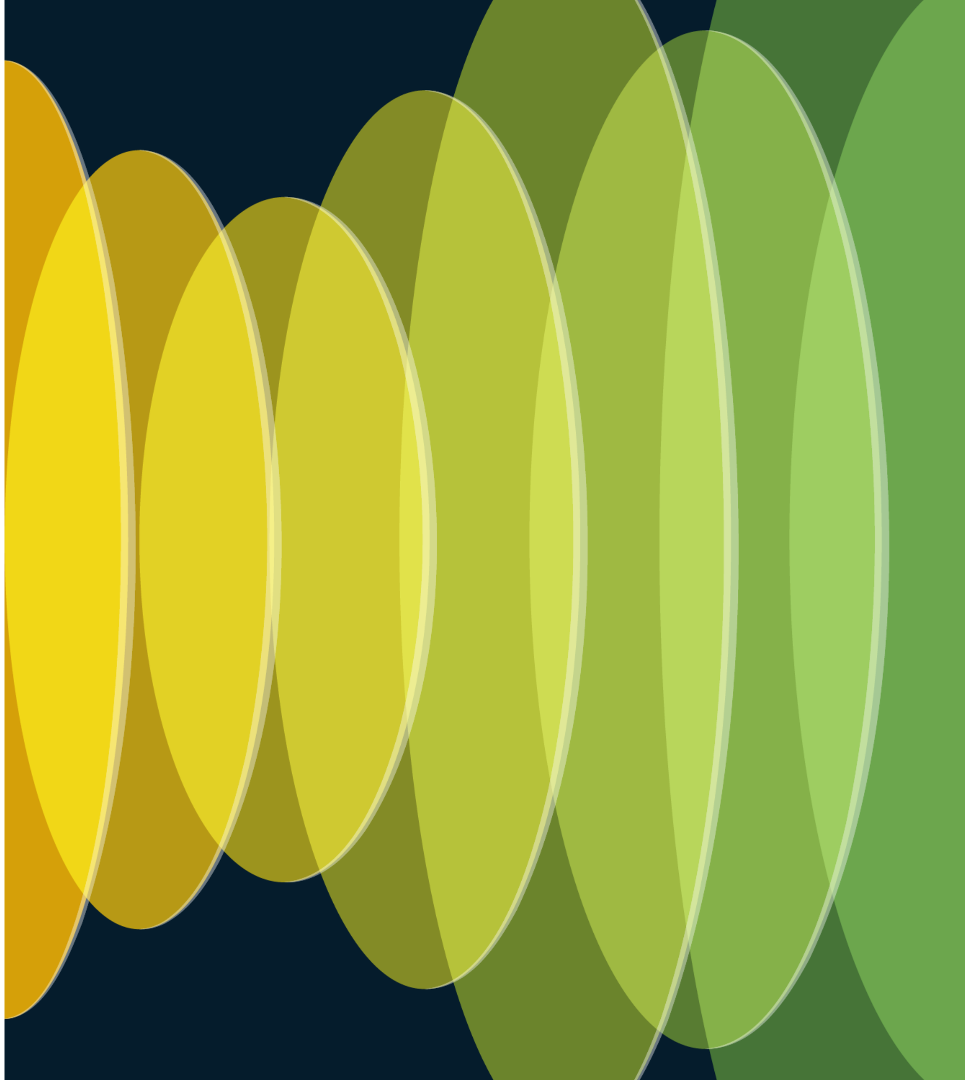
Cherry Picking



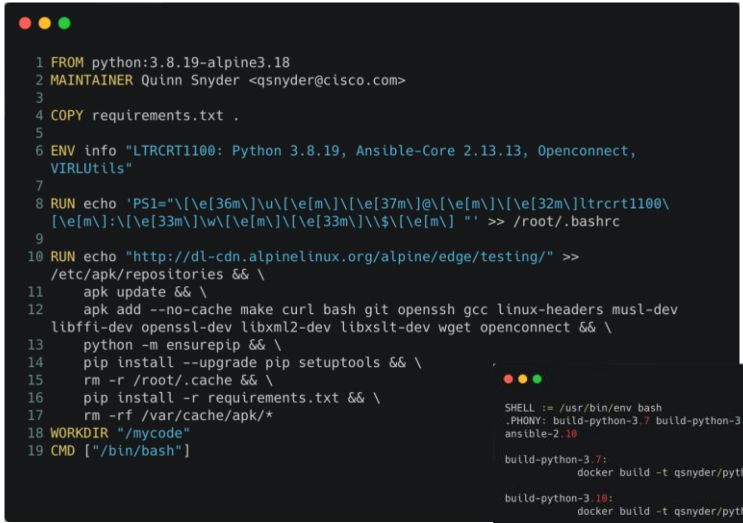
```
kiskande:03-cherry-picking/ (main) $ gi                                        [12:26:23]
```

Its OK to be \`bash\`ful

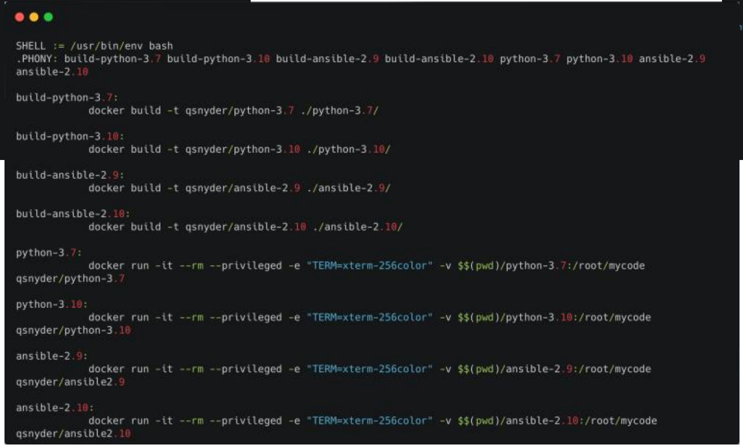
What We've Demonstrated



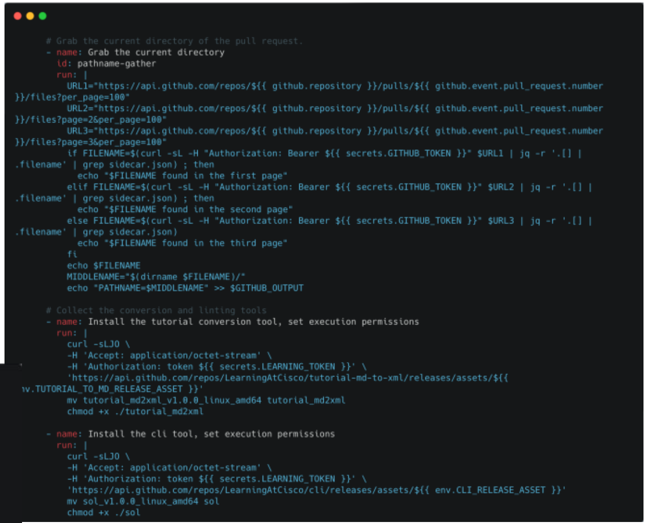
Dockerfiles, GNU `make`, SCM-based CI/CD



```
FROM python:3.8.19-alpine3.18
MAINTAINER Quinn Snyder <qsnyder@cisco.com>

COPY requirements.txt .

ENV info "LTRCRT1100: Python 3.8.19, Ansible-Core 2.13.13, Openconnect,
VIRLUtils"

RUN echo 'PS1="\[\e[36m\]\u\[\e[m\]\[\e[37m\]@\[\e[m\]\[\e[32m\]ltrcrt1100\
[\e[m\]:\[\e[33m\]\w\[\e[m\]\[\e[33m\]\\$\[\e[m\] "' >> /root/.bashrc

RUN echo "http://dl-cdn.alpinelinux.org/alpine/edge/testing/" >>
/etc/apk/repositories && \
    apk update && \
    apk add --no-cache make curl bash git openssh gcc linux-headers musl-dev
libffi-dev openssl-dev libxml2-dev libxslt-dev wget openconnect && \
    python -m ensurepip && \
    pip install --upgrade pip setuptools && \
    rm -r /root/.cache && \
    pip install -r requirements.txt && \
    rm -rf /var/cache/apk/*
WORKDIR "/mycode"
CMD ["/bin/bash"]
```



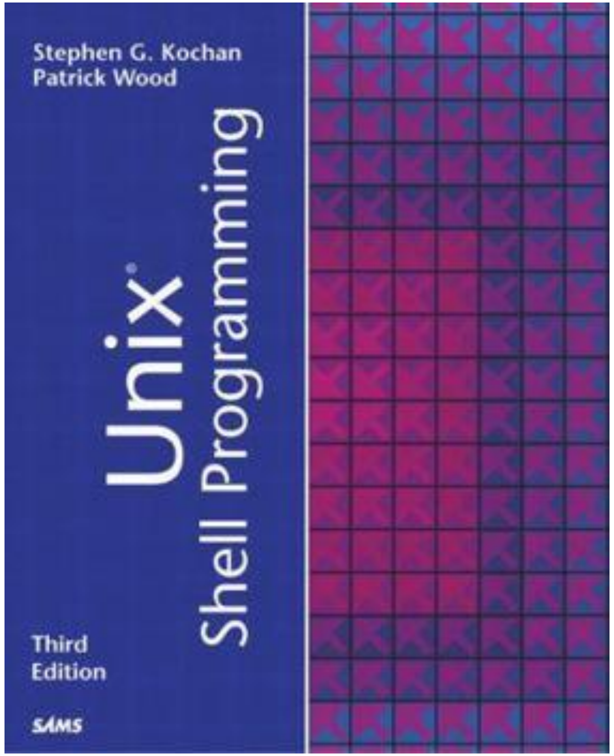
```
SHELL := /usr/bin/env bash
.PHONY: build-python-3.7 build-python-3.10 build-ansible-2.9 build-ansible-2.10 python-3.7 python-3.10 ansible-2.9
ansible-2.10

build-python-3.7:
	docker build -t qsnyder/python-3.7 ./python-3.7/

build-python-3.10:
	docker build -t qsnyder/python-3.10 ./python-3.10/

build-ansible-2.9:
	docker build -t qsnyder/ansible-2.9 ./ansible-2.9/

build-ansible-2.10:
	docker build -t qsnyder/ansible-2.10 ./ansible-2.10/

python-3.7:
	docker run -it --rm --privileged -e "TERM=xterm-256color" -v $$(pwd)/python-3.7:/root/mycode
qsnyder/python-3.7

python-3.10:
	docker run -it --rm --privileged -e "TERM=xterm-256color" -v $$(pwd)/python-3.10:/root/mycode
qsnyder/python-3.10

ansible-2.9:
	docker run -it --rm --privileged -e "TERM=xterm-256color" -v $$(pwd)/ansible-2.9:/root/mycode
qsnyder/ansible2.9

ansible-2.10:
	docker run -it --rm --privileged -e "TERM=xterm-256color" -v $$(pwd)/ansible-2.10:/root/mycode
qsnyder/ansible2.10
```



```
    # Grab the current directory of the pull request.
    - name: Grab the current directory
      id: pathname-gather
      run: |
        URL1="https://api.github.com/repos/${{ github.repository }}/pulls/${{ github.event.pull_request.number
}}/files?per_page=100"
        URL2="https://api.github.com/repos/${{ github.repository }}/pulls/${{ github.event.pull_request.number
}}/files?page=2&per_page=100"
        URL3="https://api.github.com/repos/${{ github.repository }}/pulls/${{ github.event.pull_request.number
}}/files?page=3&per_page=100"
        if FILENAME=$(curl -sL -H "Authorization: Bearer ${{ secrets.GITHUB_TOKEN }}" $URL1 | jq -r '.[] |
.filename' | grep sidecar.json) ; then
          echo "$FILENAME found in the first page"
        elif FILENAME=$(curl -sL -H "Authorization: Bearer ${{ secrets.GITHUB_TOKEN }}" $URL2 | jq -r '.[] |
.filename' | grep sidecar.json) ; then
          echo "$FILENAME found in the second page"
        else FILENAME=$(curl -sL -H "Authorization: Bearer ${{ secrets.GITHUB_TOKEN }}" $URL3 | jq -r '.[] |
.filename' | grep sidecar.json)
          echo "$FILENAME found in the third page"
        fi
        echo $FILENAME
        MIDDLENAME="$(dirname $FILENAME)/"
        echo "PATHNAME=$MIDDLENAME" >> $GITHUB_OUTPUT

    # Collect the conversion and linting tools
    - name: Install the tutorial conversion tool, set execution permissions
      run: |
        curl -sLJO \
        -H 'Accept: application/octet-stream' \
        -H 'Authorization: token ${{ secrets.LEARNING_TOKEN }}' \
        'https://api.github.com/repos/LearningAtCisco/tutorial-md-to-xml/releases/assets/${{
 v.TUTORIAL_TO_MD_RELEASE_ASSET }}'
        mv tutorial_md2xml_v1.0.0_linux_amd64 tutorial_md2xml
        chmod +x ./tutorial_md2xml

    - name: Install the cli tool, set execution permissions
      run: |
        curl -sLJO \
        -H 'Accept: application/octet-stream' \
        -H 'Authorization: token ${{ secrets.LEARNING_TOKEN }}' \
        'https://api.github.com/repos/LearningAtCisco/cli/releases/assets/${{ env.CLI_RELEASE_ASSET }}'
        mv sol_v1.0.0_linux_amd64 sol
        chmod +x ./sol
```

What’s old is new again

Who had this book at Uni?



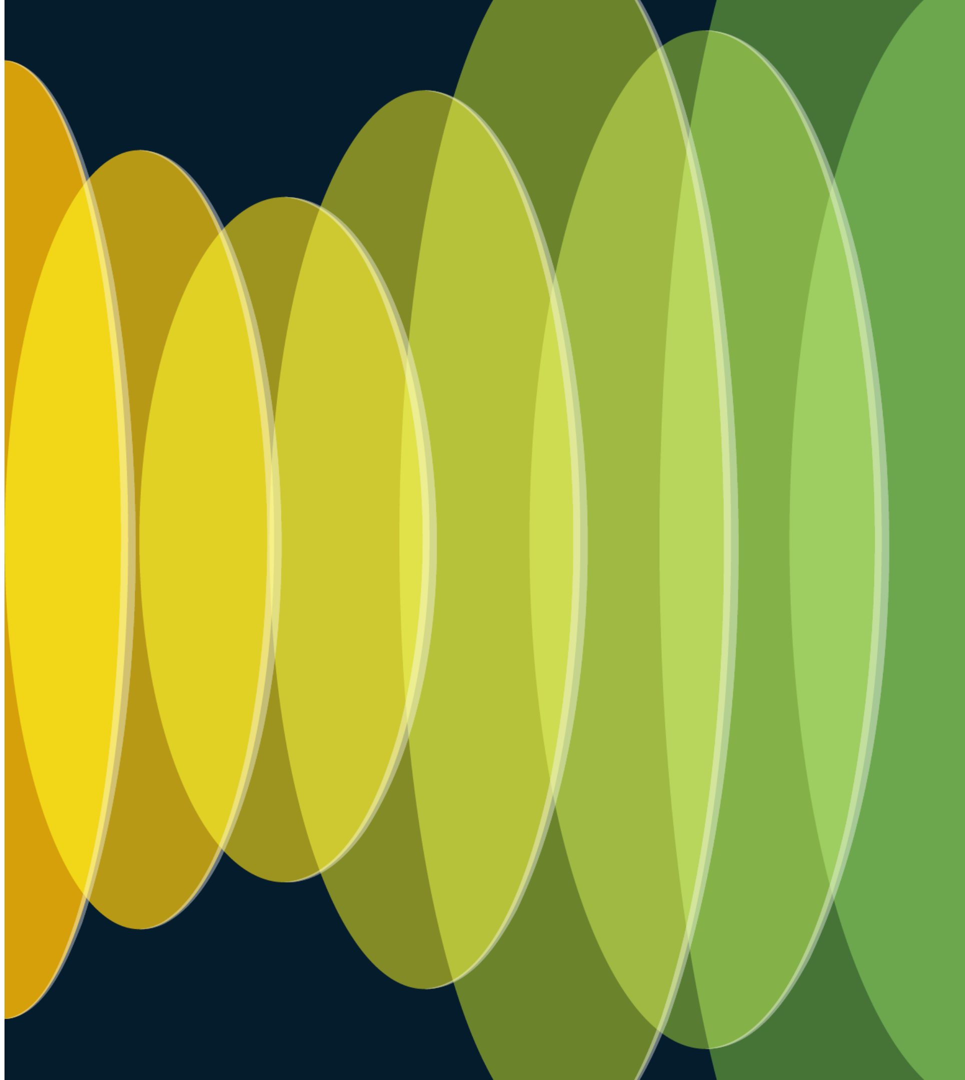
I use it more often than any book I've saved from my college education

`bash` | 36 years and still going strong!

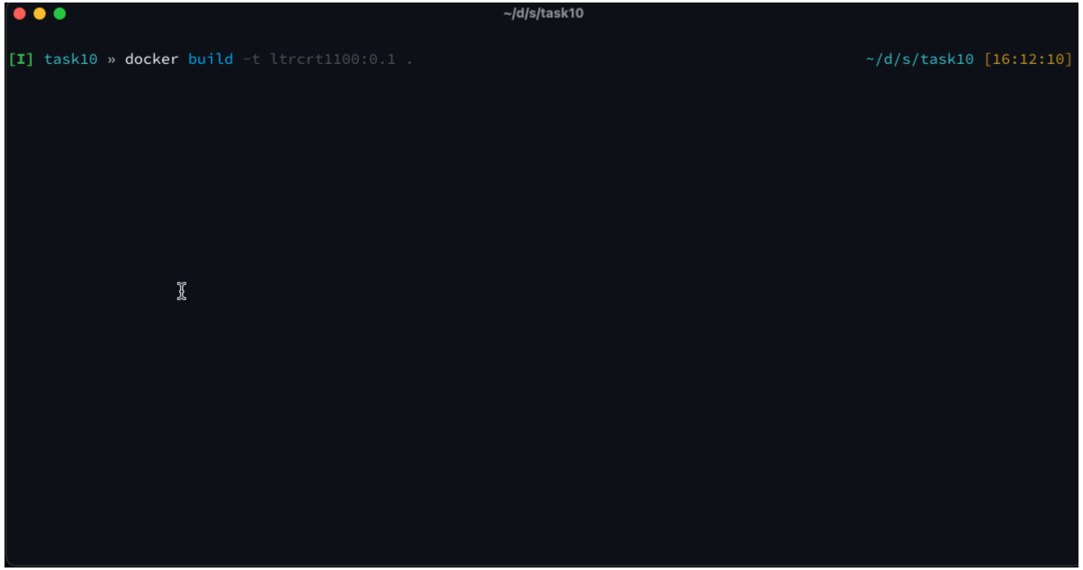
- Everything entered at the Linux CLI* can become part of a shell script or CI/CD pipeline
  - Yes, its all bash scripts and YAML
  - This is “imperative language” which is still a thing in 2024
- Scripts can still execute other higher order processes
  - Python, Ansible, Terraform, etc
- Dockerfiles, CI/CD pipeline declarations (Gitlab, GitHub Actions)
  - May have specific runners or actions to hide abstraction
  - Everything is running some sort of Linux* container

*Assuming POSIX-shells and so on…

Practical Applications



Bash can build locally



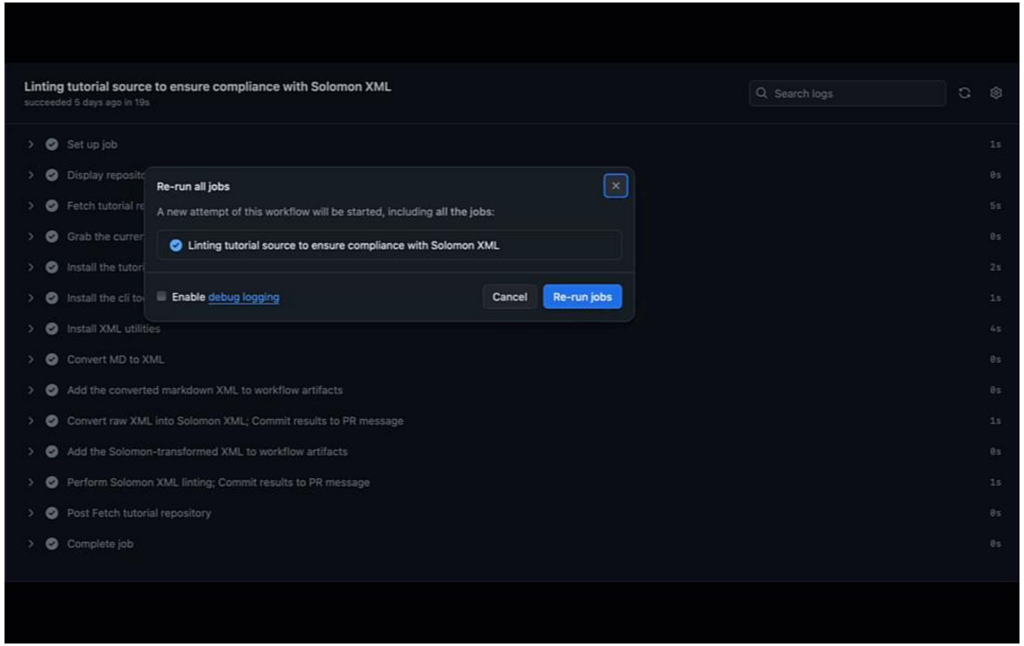
```
~/d/s/task10
[I] task10 » docker build -t ltrcrt1100:0.1 .                                    ~/d/s/task10 [16:12:10]
```

Whether using native Docker CLI, or `makefile` with targets, commands run within container are based on Linux CLI

Bash can build in the cloud

GitHub Actions are container runners of various OS-types.

Some pre-built functions exist, but bespoke scripts may be needed, even if calling Python/Go/etc.



Linting tutorial source to ensure compliance with Solomon XML
succeeded 5 days ago in 19s

Set up job	1s
Display reposito	0s
Fetch tutorial re	5s
Grab the curren	0s
Install the tutori	2s
Install the cli to	1s
Install XML utilities	4s
Convert MD to XML	0s
Add the converted markdown XML to workflow artifacts	0s
Convert raw XML into Solomon XML; Commit results to PR message	1s
Add the Solomon-transformed XML to workflow artifacts	0s
Perform Solomon XML linting; Commit results to PR message	1s
Post Fetch tutorial repository	0s
Complete job	0s

Re-run all jobs

A new attempt of this workflow will be started, including all the jobs:

Linting tutorial source to ensure compliance with Solomon XML

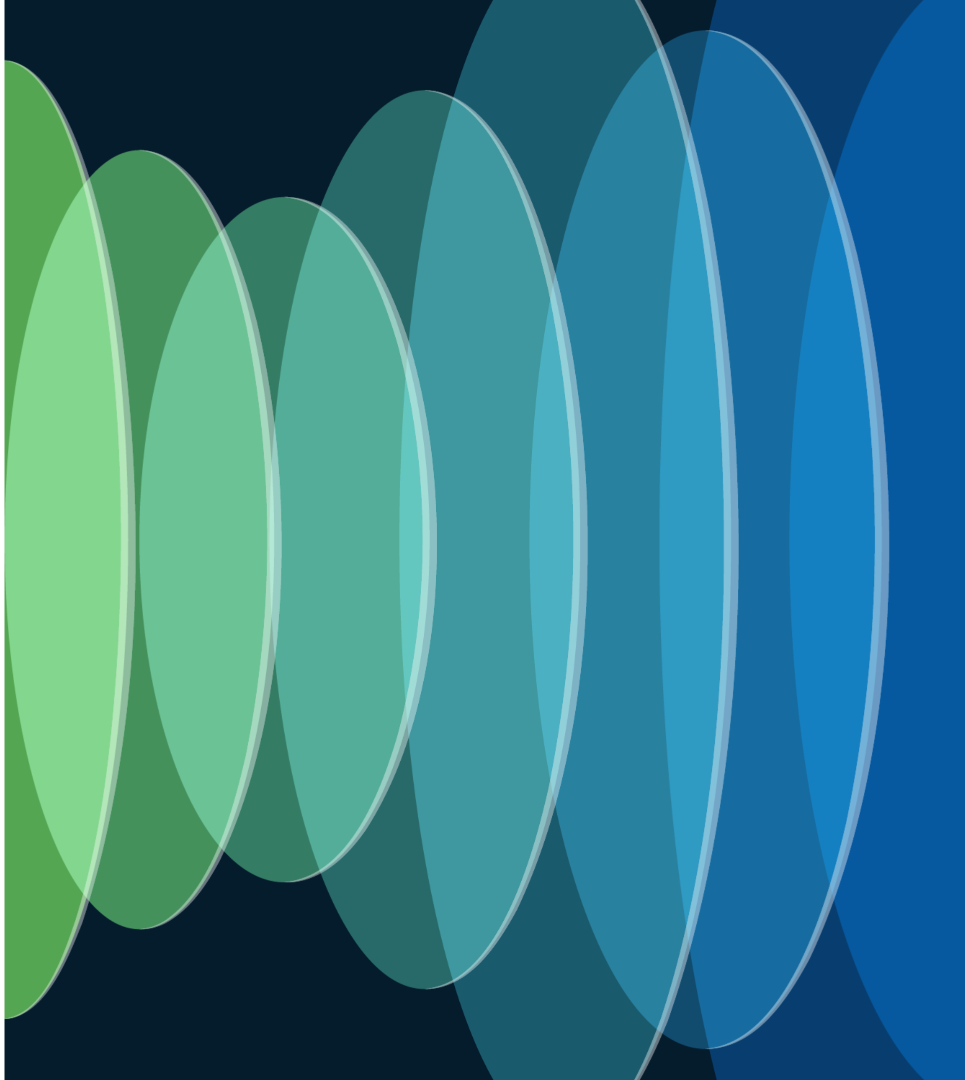
Enable debug logging

"When I die, I don't care about clearing my browser history. I want someone to delete my ~/dev folder to hide the heinous things I did with bash scripts"

~ Some Bearded Jabroni

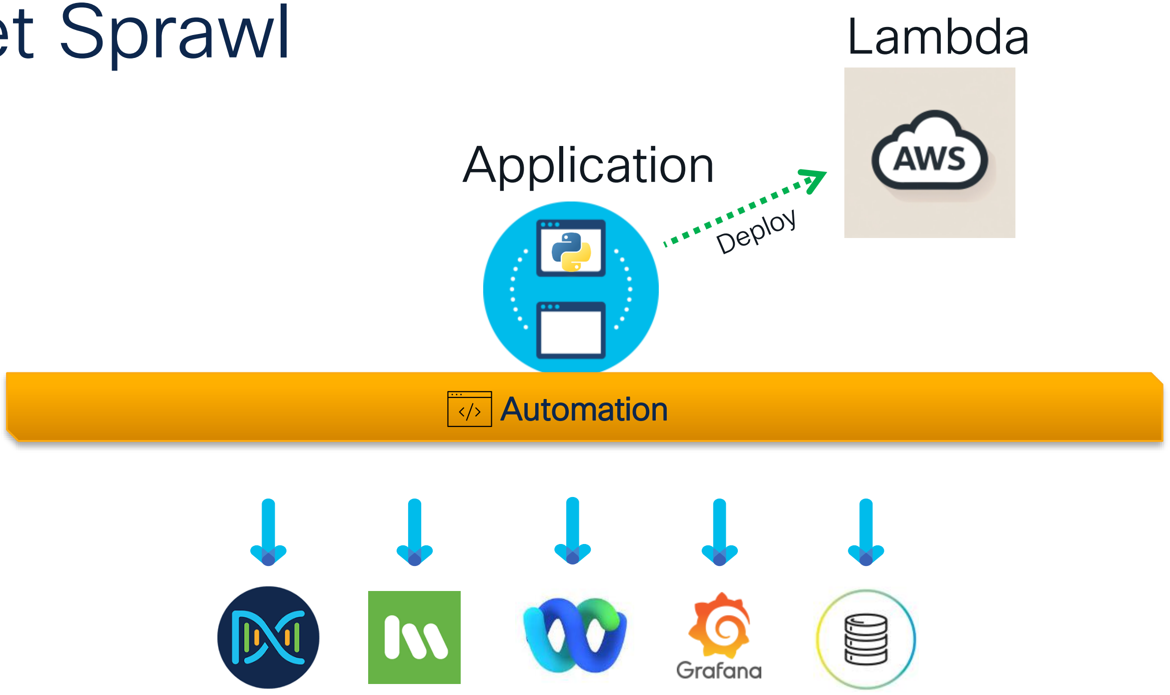
Locking the Vault

What we've been talking to you about



cisco Live!

Secret Sprawl



~7 Different API Tokens | SSH Keys | Auth Methods

Vault

HashiCorp Vault - Secrets SSOT

Single Source of Truth for all you Secretes

- API Tokens
- SSH Keys
- Basic Auth - Username/Password
- DB Creds

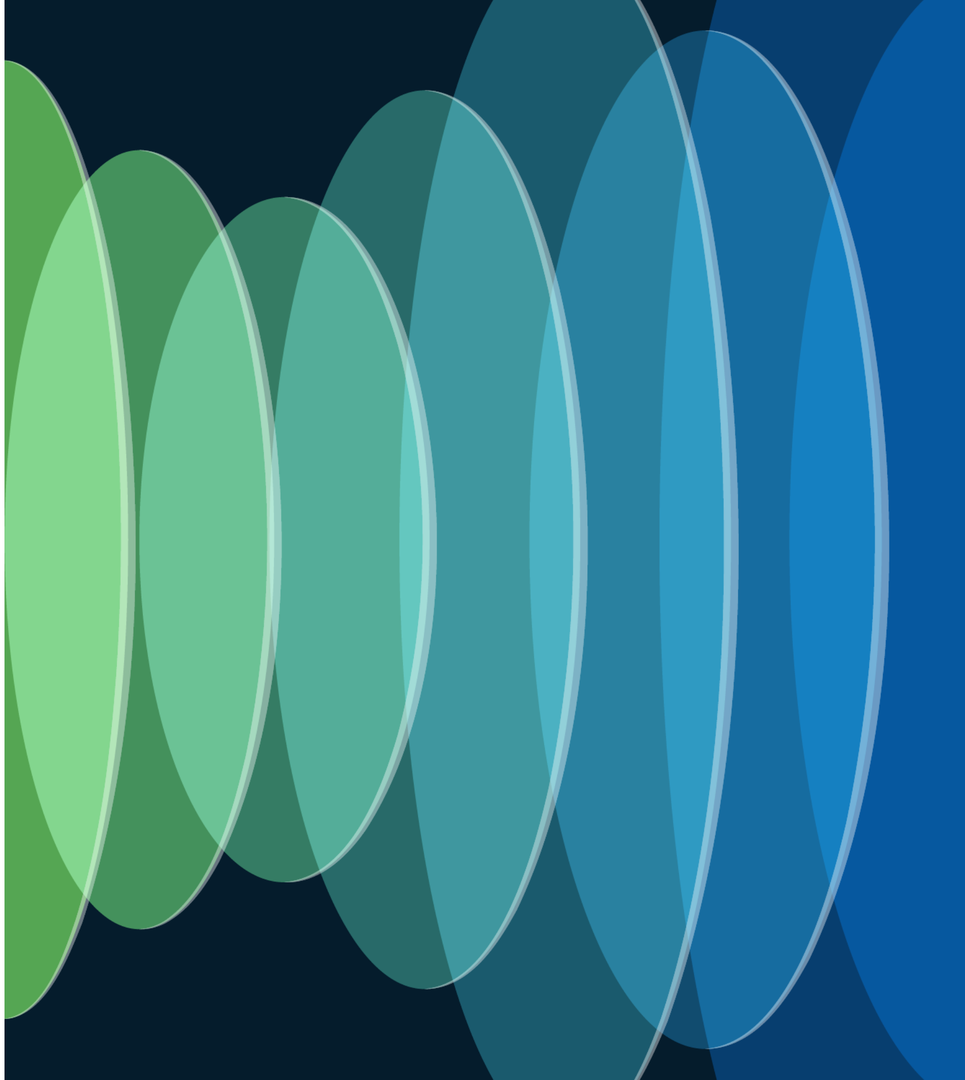
Granular Access Control (ACLs)

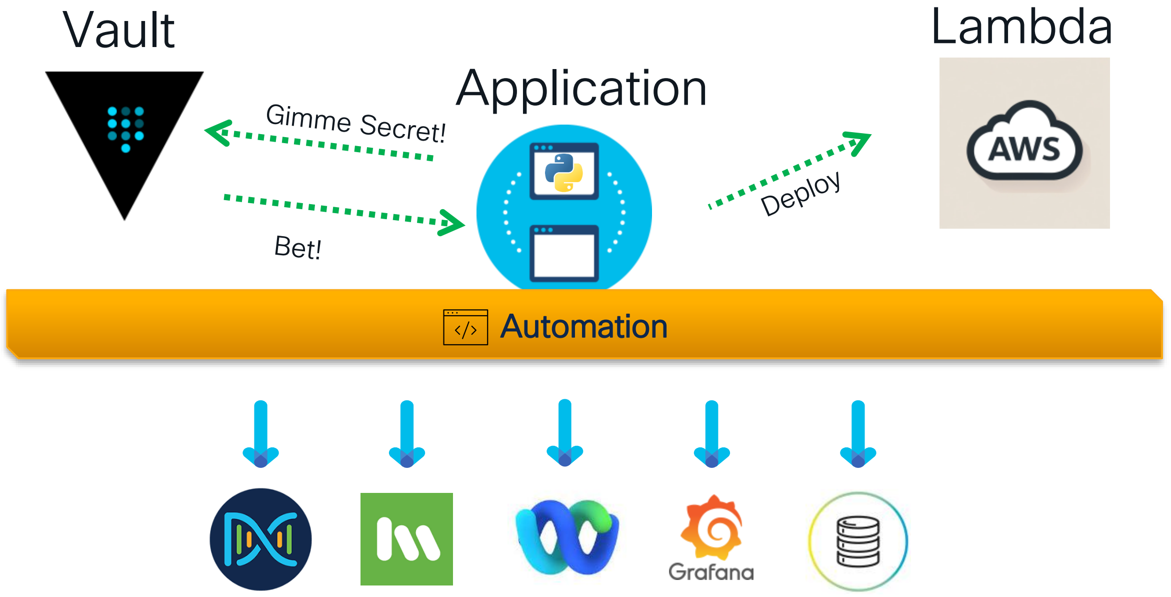
- Define App Rules
- Apps granted access to Mount/Path
- Paths Contain Creds

Vault APIs FTW

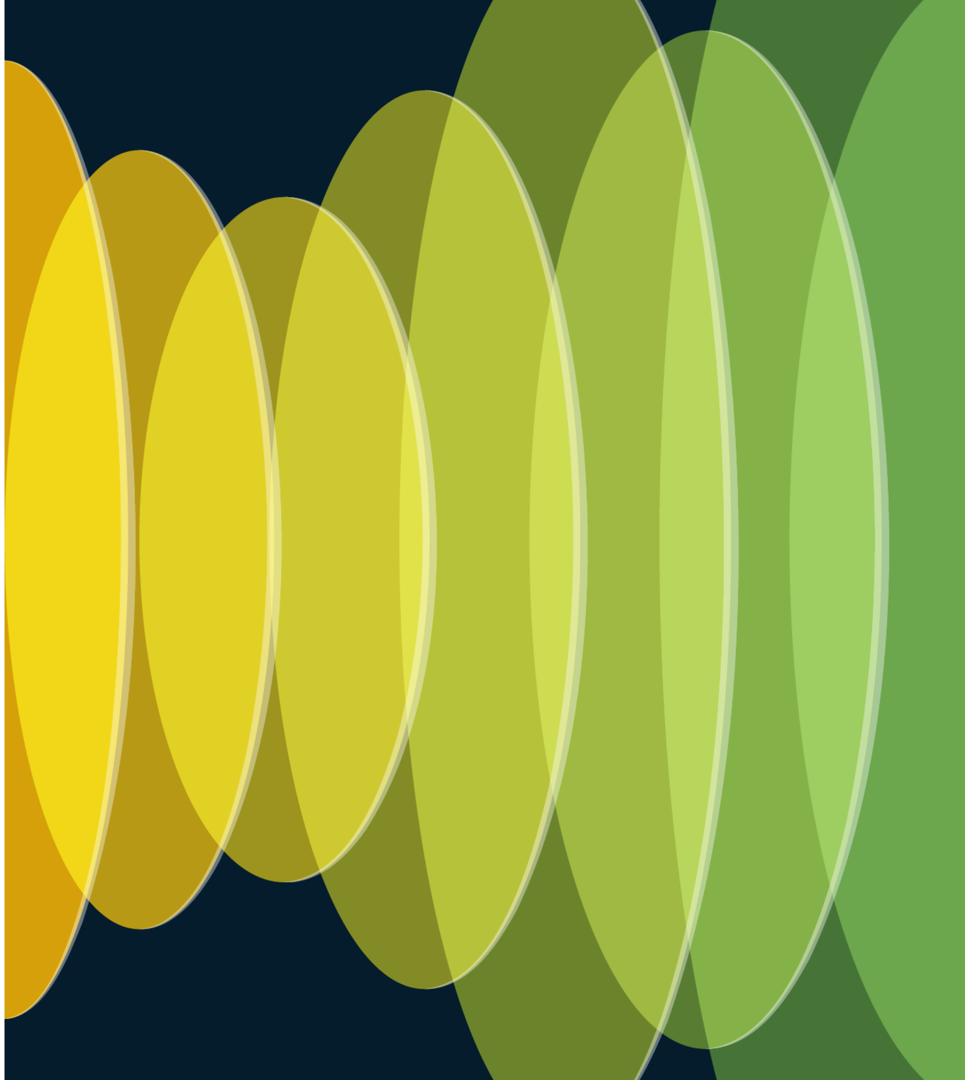
- API access to your creds
- Seamless integration with your App
- Bonus Points - HVAC SDK!!

Real World



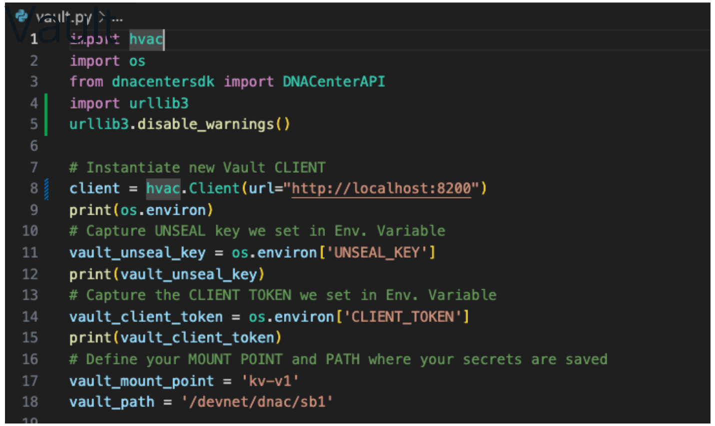

Vault
Application
Lambda
Gimme Secret!
Bet!
Deploy
AWS
Automation
Grafana

Let’s try it!



VAULT HVAC SDK + DNACSDK = <3

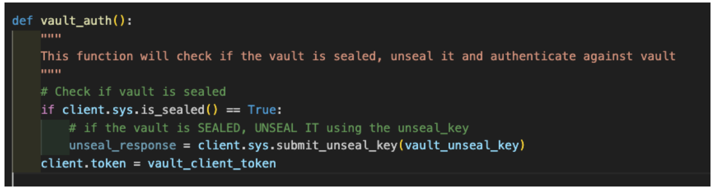
1. Instantiate Vault



vault.py > ...

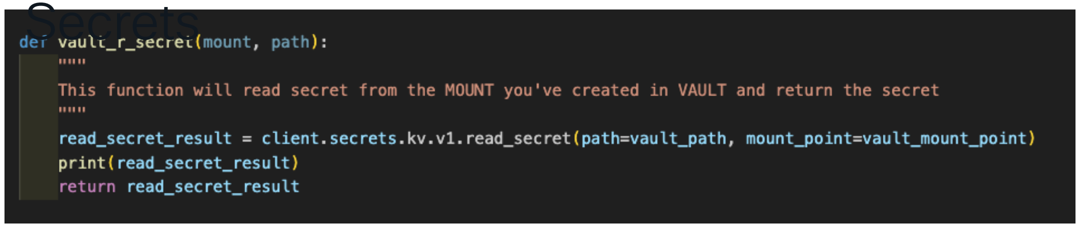
```python
import hvac
import os
from dnacentersdk import DNACenterAPI
import urllib3
urllib3.disable_warnings()

# Instantiate new Vault CLIENT
client = hvac.Client(url="http://localhost:8200")
print(os.environ)
# Capture UNSEAL key we set in Env. Variable
vault_unseal_key = os.environ['UNSEAL_KEY']
print(vault_unseal_key)
# Capture the CLIENT TOKEN we set in Env. Variable
vault_client_token = os.environ['CLIENT_TOKEN']
print(vault_client_token)
# Define your MOUNT POINT and PATH where your secrets are saved
vault_mount_point = 'kv-v1'
vault_path = '/devnet/dnac/sb1'
```

2. Unseal Vault



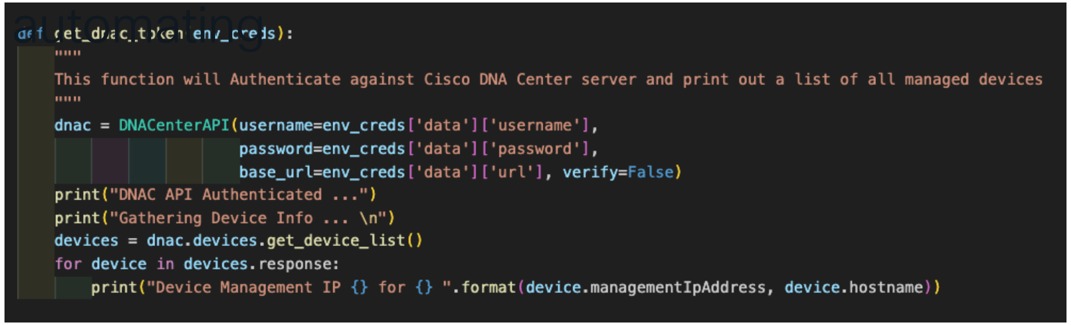
```python
def vault_auth():
    """
    This function will check if the vault is sealed, unseal it and authenticate against vault
    """
    # Check if vault is sealed
    if client.sys.is_sealed() == True:
        # if the vault is SEALED, UNSEAL IT using the unseal_key
        unseal_response = client.sys.submit_unseal_key(vault_unseal_key)
    client.token = vault_client_token
```

3. Read the Secrets



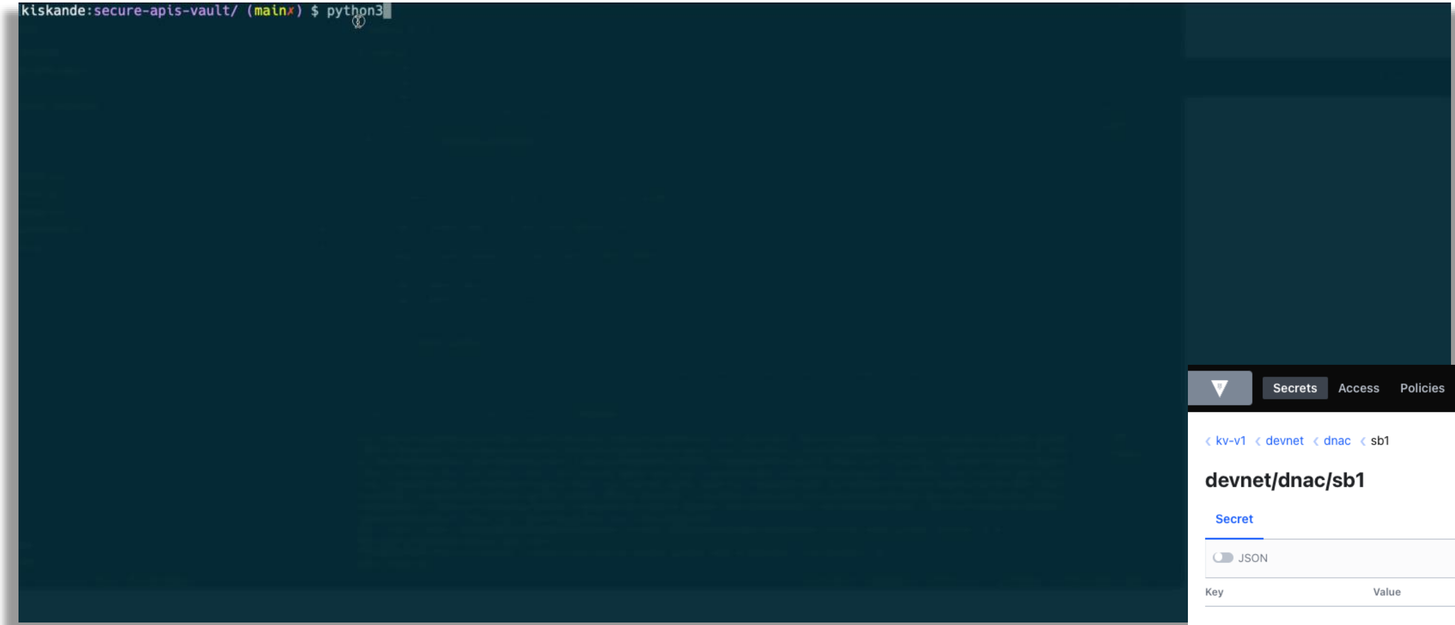
```python
def vault_r_secret(mount, path):
    """
    This function will read secret from the MOUNT you've created in VAULT and return the secret
    """
    read_secret_result = client.secrets.kv.v1.read_secret(path=vault_path, mount_point=vault_mount_point)
    print(read_secret_result)
    return read_secret_result
```

4. Start automating

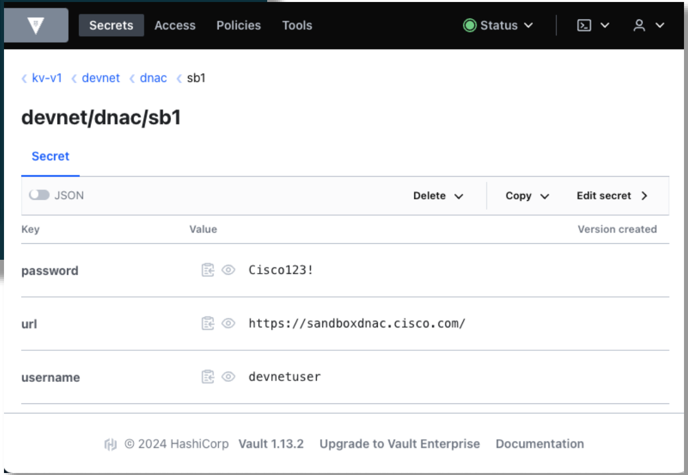


```python
def get_dnac_token(env_creds):
    """
    This function will Authenticate against Cisco DNA Center server and print out a list of all managed devices
    """
    dnac = DNACenterAPI(username=env_creds['data']['username'],
                        password=env_creds['data']['password'],
                        base_url=env_creds['data']['url'], verify=False)
    print("DNAC API Authenticated ...")
    print("Gathering Device Info ... \n")
    devices = dnac.devices.get_device_list()
    for device in devices.response:
        print("Device Management IP {} for {} ".format(device.managementIpAddress, device.hostname))
```

VAULT HVAC SDK + DNACSDK = <3



```
kiskande:secure-apis-vault/ (main✗) $ python3
```



kv-v1 ‹ devnet ‹ dnac ‹ sb1

devnet/dnac/sb1

Secret

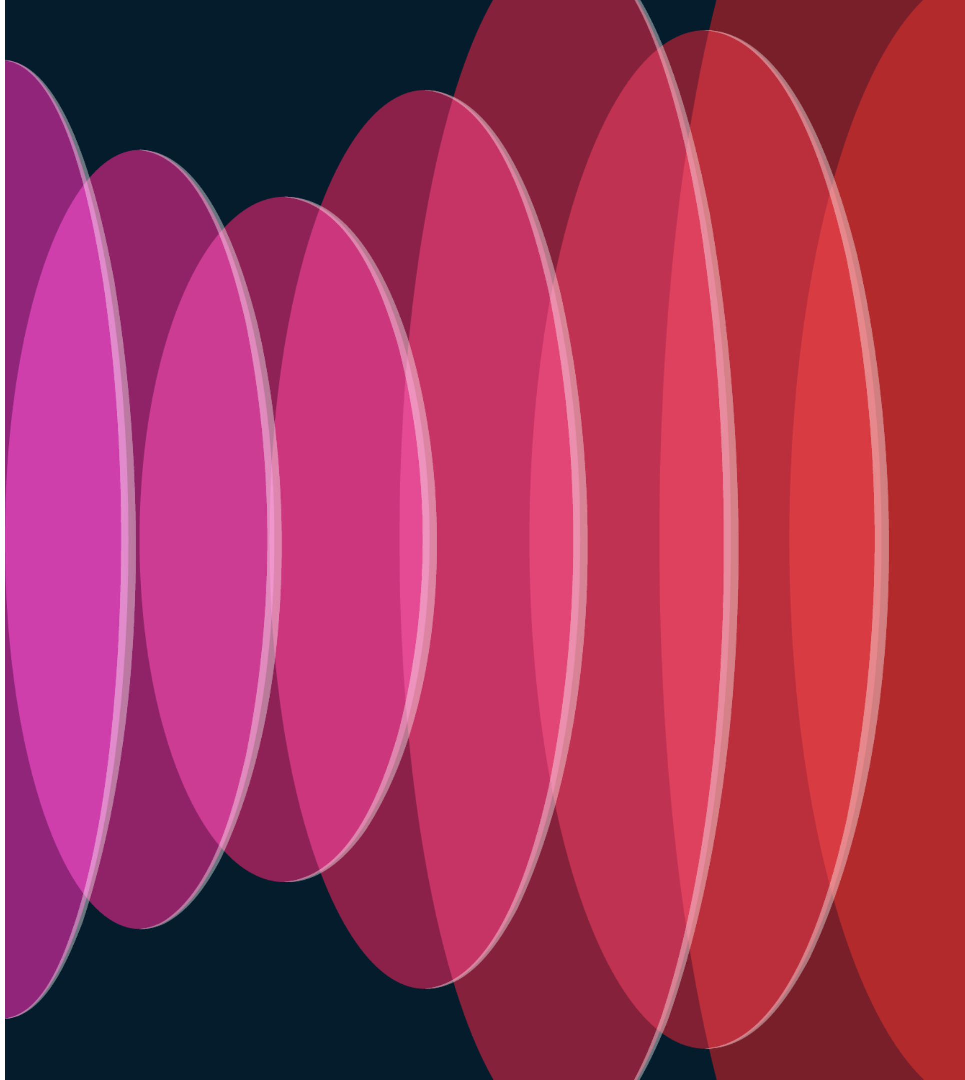
JSON Delete Copy Edit secret

Key	Value	Version created
password	Cisco123!	
url	https://sandboxdnac.cisco.com/	
username	devnetuser	

© 2024 HashiCorp Vault 1.13.2 Upgrade to Vault Enterprise Documentation

Codifying your Infra

What I've Droned on About



cisco Live!

Mostly About IaC in Single Context

The bridge to possible

Standardize Your Network

Building Roles for Device Compliance Using Ansible

Quinn Snyd
@qsnyder
DEVWKS-2252

The bridge to possible

Prepare for the Enterprise Automation (ENAUTO) Certification with Real-Life Applications

George Koukis, Leader Exam Progr
Quinn Snyder, Senior Technical Adv
BRKCRT-2014

```
- name: Testing our fact-finding mission
  hosts: all
  gather_facts: no

  tasks:

   - name: Gather IOSXE Facts
     cisco.ios.ios_facts:
       gather_subset:
       - config
     when: ansible_network_os == 'cisco.ios.ios'

   - name: Gather NXOS Facts
     cisco.nxos.nxos_facts:
       gather_subset:
       - config
     when: ansible_network_os == 'cisco.nxos.nxos'
```

The bridge to possible

Getting Started with IaC and Terraform

Have No Fear, HCL is

Quinn Snyder
Developer Advocate, Cisc
DEVLIT-2785

Typical Terraform Workflow

Binary Unzip/Install
Add the binary to your $PATH
Creation of TF Configuration in HCL
Create the configuration for the endpoint resource along with any variables
Terraform Init
Reads the configuration, downloads and initializes the provider for this project
Terraform Plan
Determines the delta and steps to apply desired configuration to the resource
Terraform Apply
Applies the steps created from running the plan command
Terraform Destroy
Removes the configured state as

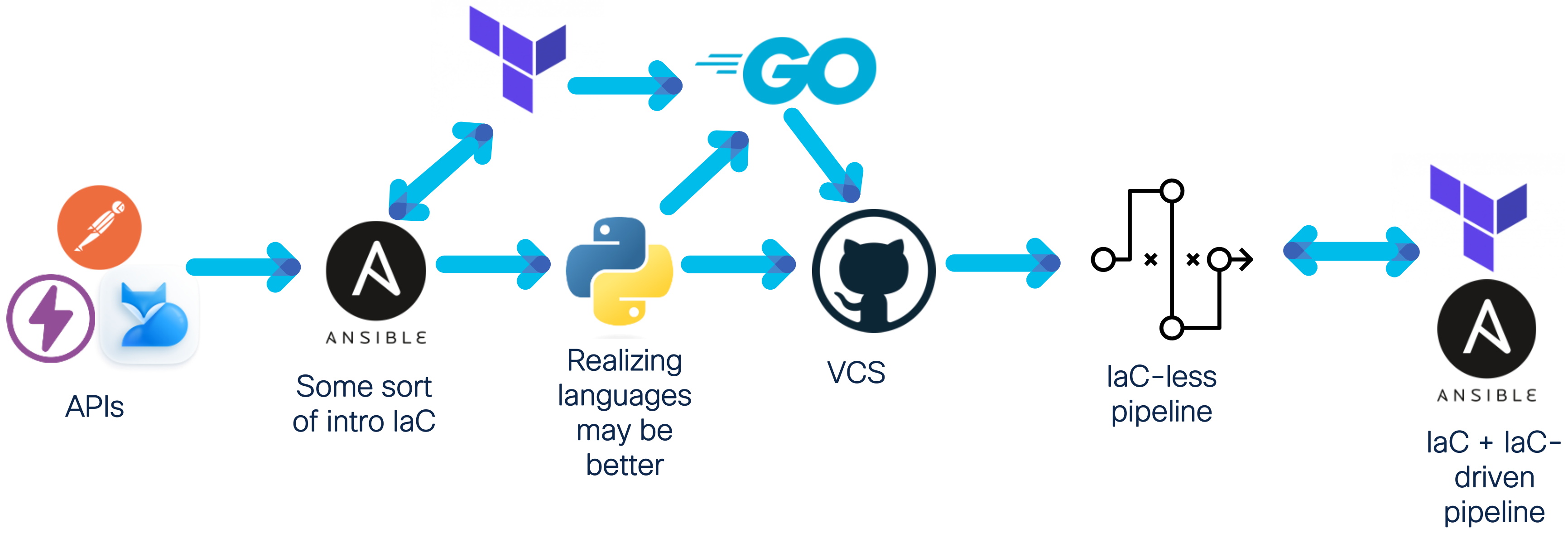
HCL Basics

- All HCL plans of similar structure
- Possible to "read-in" for reuse without config
- Note the dotted notation nesting for relationships
- Sometimes variables are exposed without being declared as part of the provider

```
# Define an MSO Tenant Resource.
data "mso_tenant" "tenant_obj" {
    name         = var.tenant
    display_name = var.tenant
}

# Define an MSO Schema Resource.
resource "mso_schema" "schema_obj" {
    template_name = "Template1"
    name          = var.schema
    tenant_id     = data.mso_tenant.tenant_obj.id
}

# Define an MSO Schema VRF Resource.
resource "mso_schema_template_vrf" "vrf_obj" {
    schema_id     = mso_schema.schema_obj.id
    template      = mso_schema.schema_obj.template_name
    name          = var.vrf
    display_name  = var.vrf
}

# Define an MSO Schema BD Resource.
resource "mso_schema_template_bd" "bd_obj" {
    schema_id              = mso_schema.schema_obj.id
    template_name          = mso_schema.schema_obj.template_name
    name                   = var.bd
    display_name           = var.bd
    vrf_name               = mso_schema_template_vrf.vrf_obj.name
    layer2_unknown_unicast = "proxy"
    layer2_stretch         = true
}
```

#CiscoLive DEVLIT-2785 © 2022 Cisco and/or its affiliates. All rights reserved. Cisco Public 7

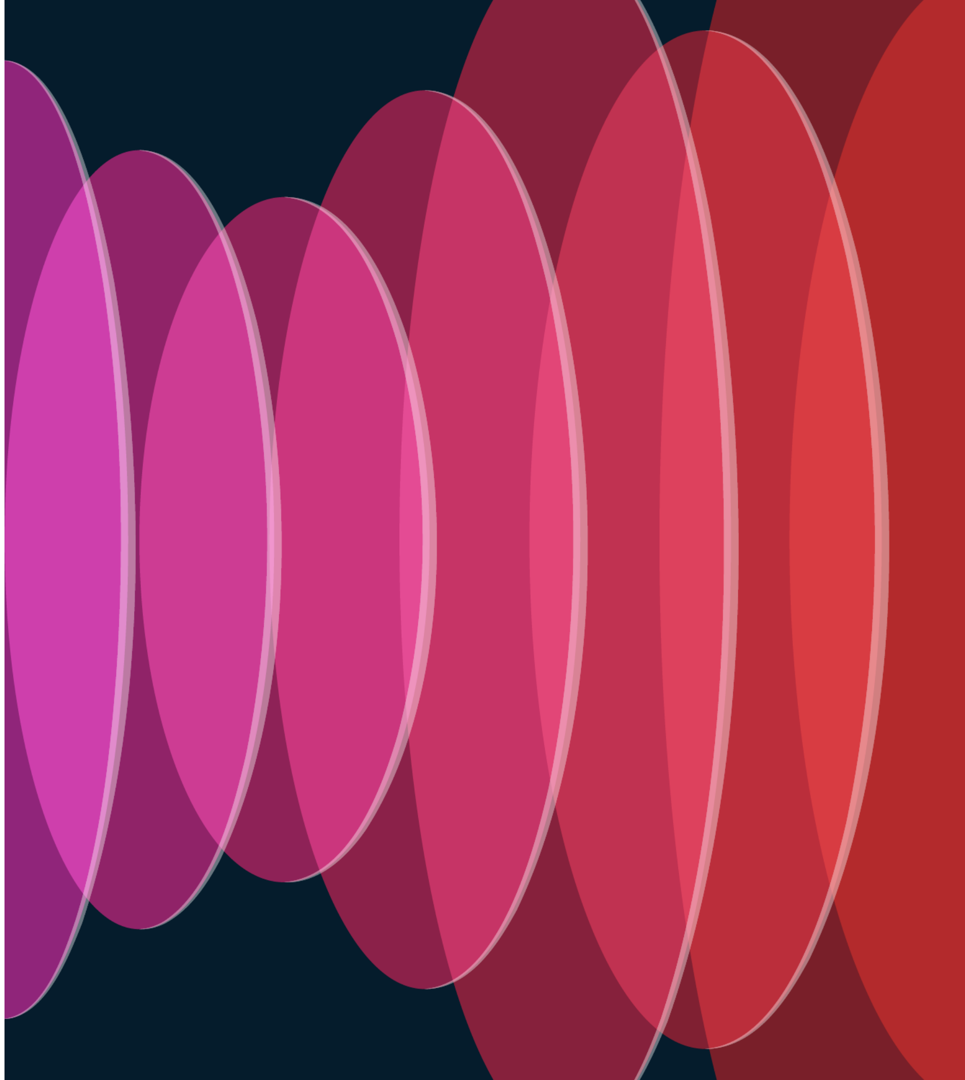
And I talk about an automation journey like this...



And IaC is GREAT for Cloud

```
[I] testing »                                           ~/D/c/e/a/testing [15:27:16]
```

But How Does That Apply in Practice?



We Can Also Convert Click-ops into IaC

```
~/D/c/e/a/testing

[I] testing »                                    ~/D/c/e/a/testing [16:07:45]
```

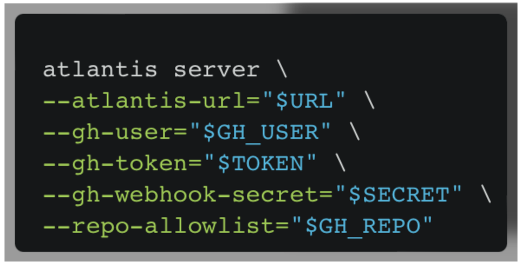
Terraformer:

- https://github.com/GoogleCloudPlatform/terraformer

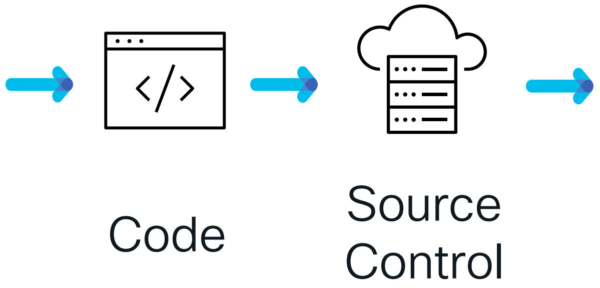
Azure Export for Terraform:

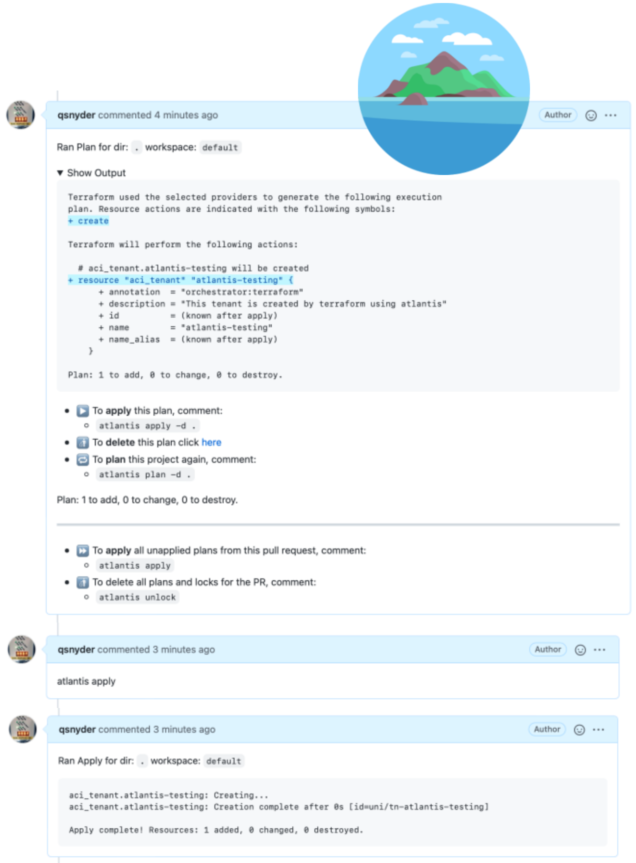
- https://github.com/Azure/aztfexport

IaC Enables Easier Pipelines*



```
atlantis server \
--atlantis-url="$URL" \
--gh-user="$GH_USER" \
--gh-token="$TOKEN" \
--gh-webhook-secret="$SECRET" \
--repo-allowlist="$GH_REPO"
```





qsnyder commented 4 minutes ago · Author

Ran Plan for dir: `.` workspace: `default`

▼ Show Output

```
Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

  # aci_tenant.atlantis-testing will be created
+ resource "aci_tenant" "atlantis-testing" {
      + annotation  = "orchestrator:terraform"
      + description = "This tenant is created by terraform using atlantis"
      + id          = (known after apply)
      + name        = "atlantis-testing"
      + name_alias  = (known after apply)
    }

Plan: 1 to add, 0 to change, 0 to destroy.
```

- To **apply** this plan, comment:
  - `atlantis apply -d .`
- To **delete** this plan click here
- To **plan** this project again, comment:
  - `atlantis plan -d .`

Plan: 1 to add, 0 to change, 0 to destroy.

- To **apply** all unapplied plans from this pull request, comment:
  - `atlantis apply`
- To delete all plans and locks for the PR, comment:
  - `atlantis unlock`

qsnyder commented 3 minutes ago · Author

atlantis apply

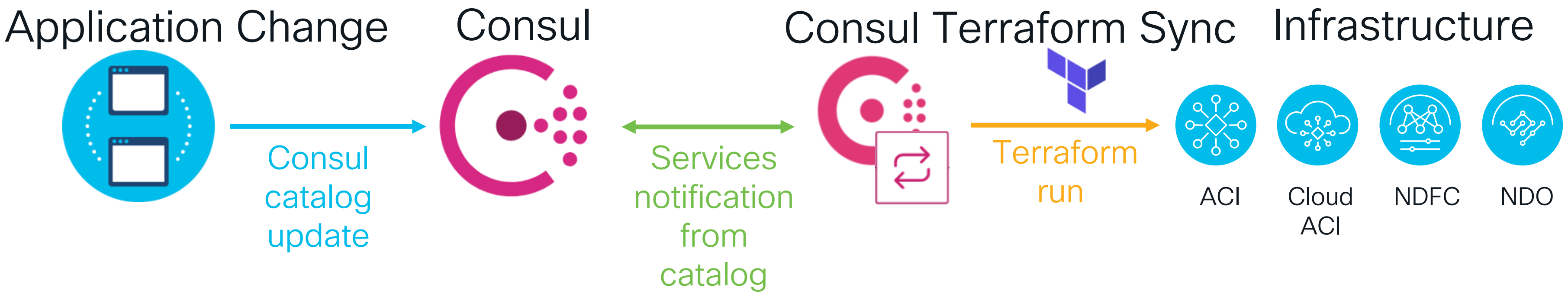
qsnyder commented 3 minutes ago · Author

Ran Apply for dir: `.` workspace: `default`

```
aci_tenant.atlantis-testing: Creating...
aci_tenant.atlantis-testing: Creation complete after 0s [id=uni/tn-atlantis-testing]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

*Please don't tell `bash` I'm cheating...

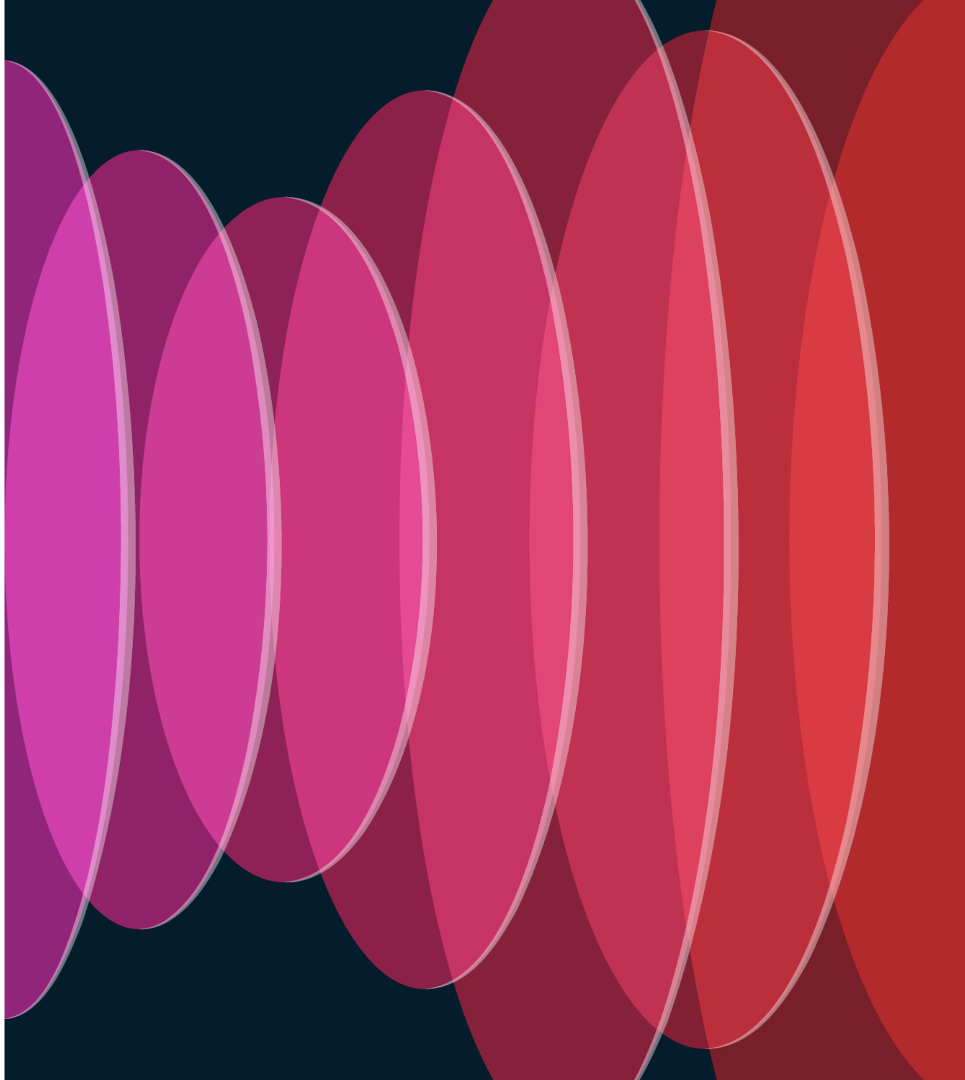
IaC Enables VCS-less Pipelines*



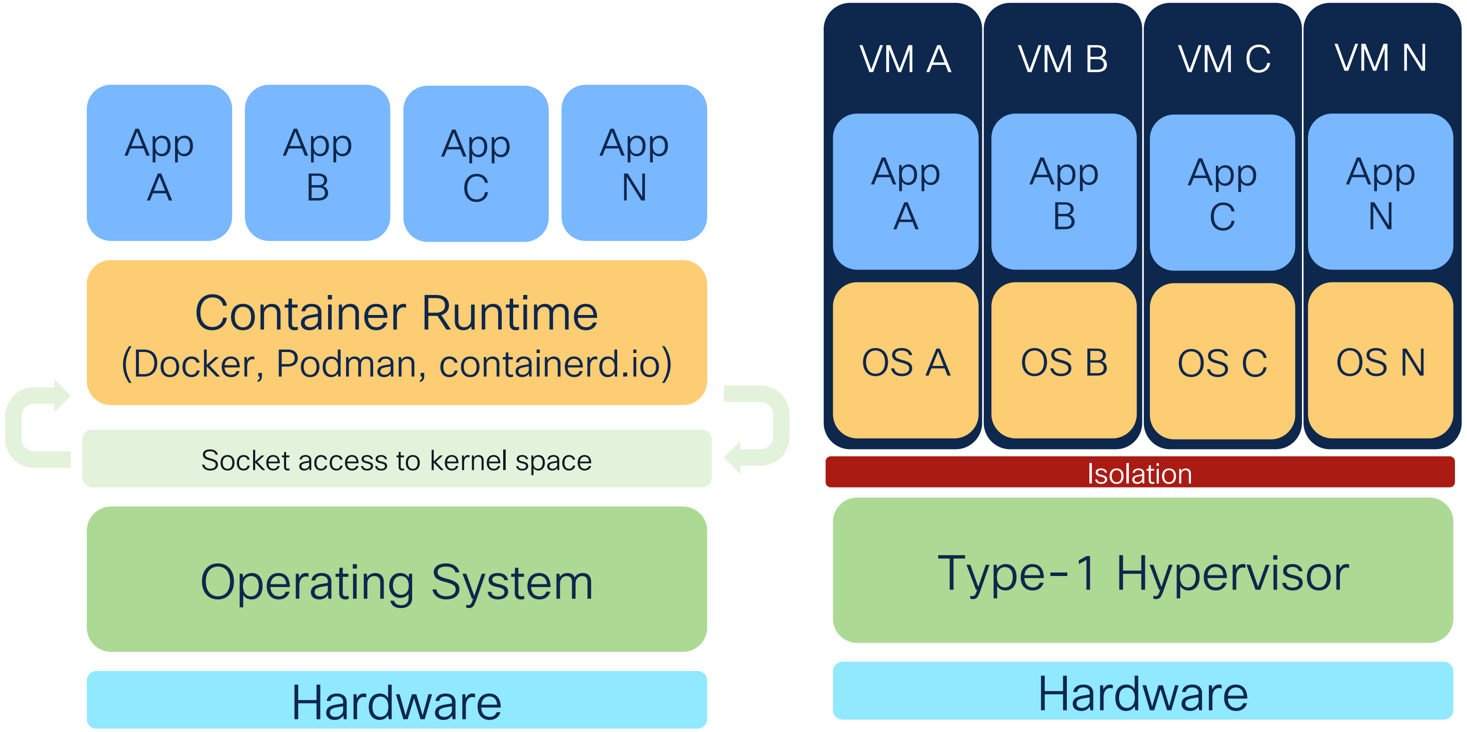
*Again, don't tell `bash` I'm cheating…

Containerize This!

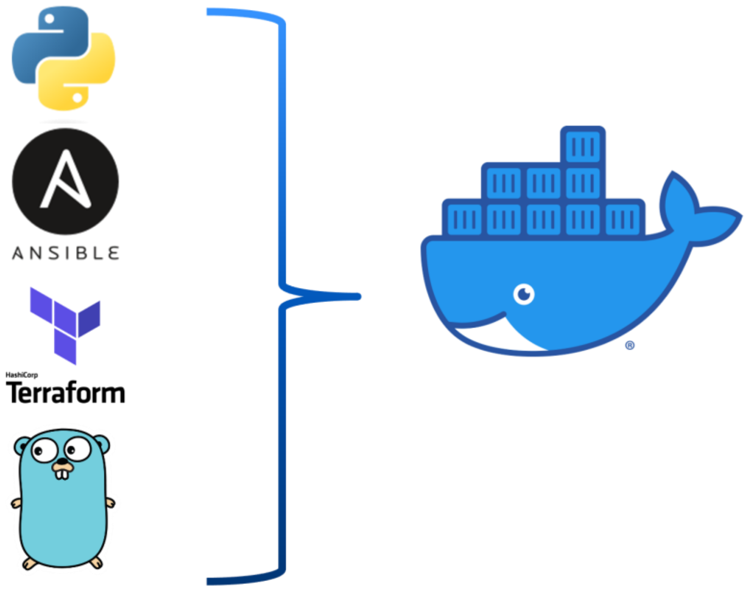
I’ve Said This Previously



What Makes Containers Unique



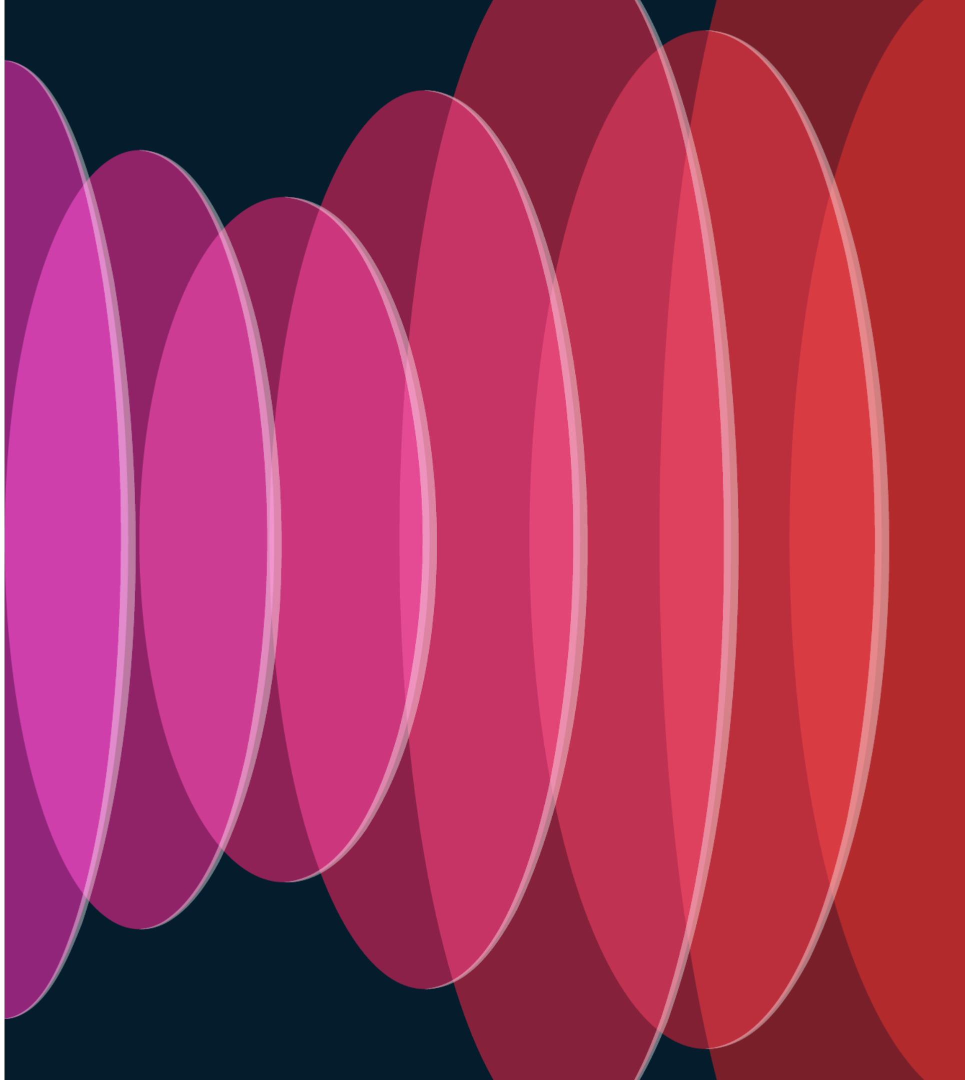
Package Everything Up and Ship It



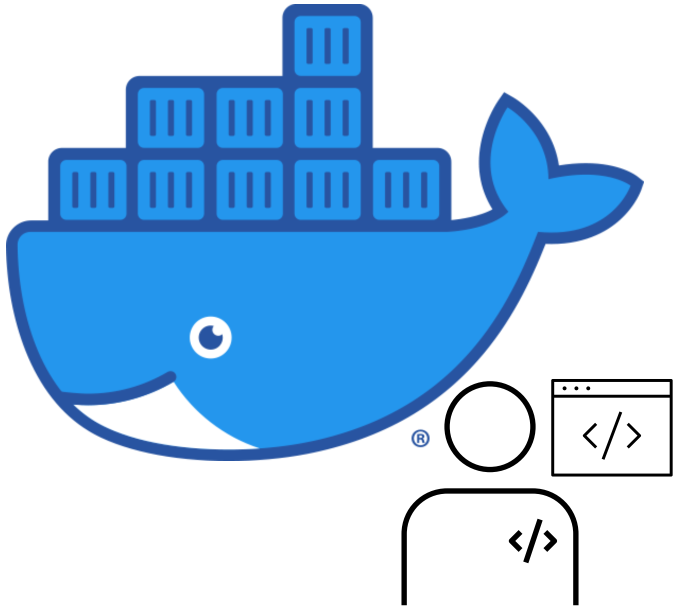
Containers create uniform package runtime environment (like Java was supposed to do, but better)

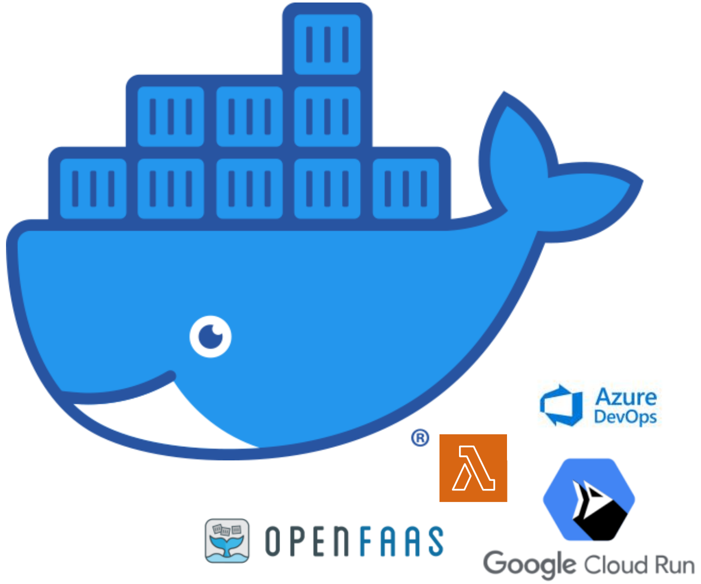
Most* abstractions survive across OS and cloud providers

Practically Speaking Though…

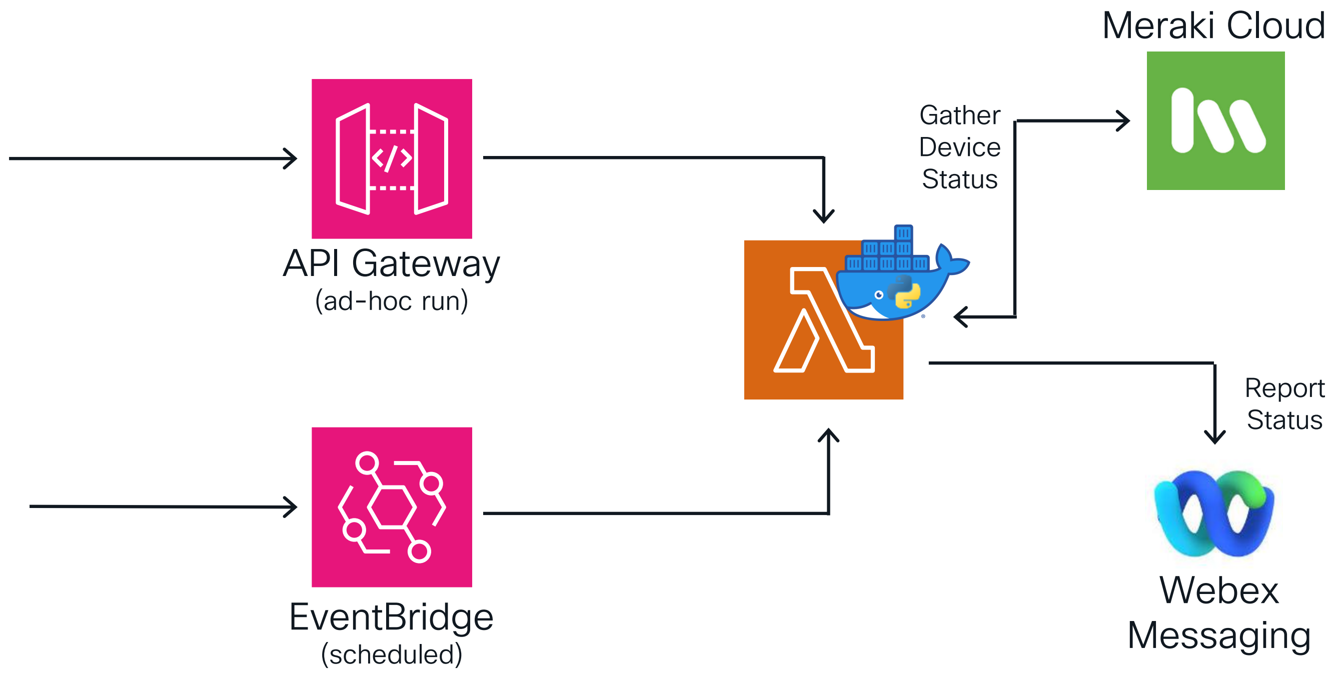


Containers Can Do Both!





Level Up Containers in the Cloud!



Wrapping it all up

Its All Connected

Vault

One More

A&O

Infra as Code

Bash

Cloud

CI/CD

Containers

Git

Complete Your Session Evaluations

Complete a minimum of 4 session surveys and the Overall Event Survey to be entered in a drawing to **win 1 of 5 full conference passes** to Cisco Live 2025.

Earn 100 points per survey completed and compete on the Cisco Live Challenge leaderboard.

Level up and earn **exclusive prizes!**

Complete your surveys in the **Cisco Live mobile app.**

Continue your education

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

The bridge to possible

Thank you

#CiscoLive