

The background features a vibrant, multi-colored abstract design. On the left, there are horizontal, wavy bands of color in shades of red, orange, yellow, and green. On the right, a bright white light source emits a series of sharp, triangular rays in various shades of blue and cyan, creating a sunburst effect that spreads across the right half of the image.

cisco *Live!*

Let's go

#CiscoLiveAPJC



The bridge to possible

Extending CML

Terraforming the Lost City

Quinn Snyder | Technical Advocate, Learning and Certifications

@qsnyder

DEVNET-1174

CISCO *Live!*

#CiscoLiveAPJC

Cisco Webex App

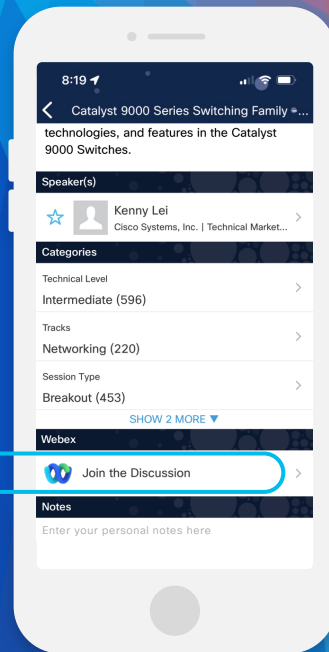
Questions?

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 December 22, 2023.

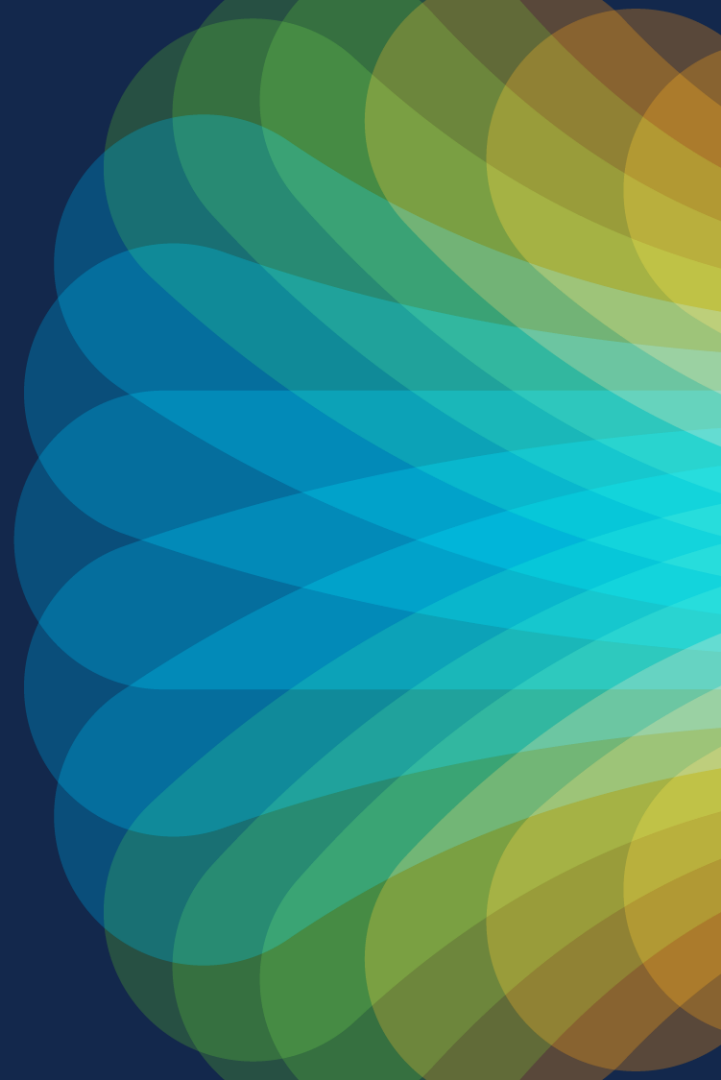


<https://ciscolive.ciscoevents.com/ciscolivebot/#DEVNET-1174>

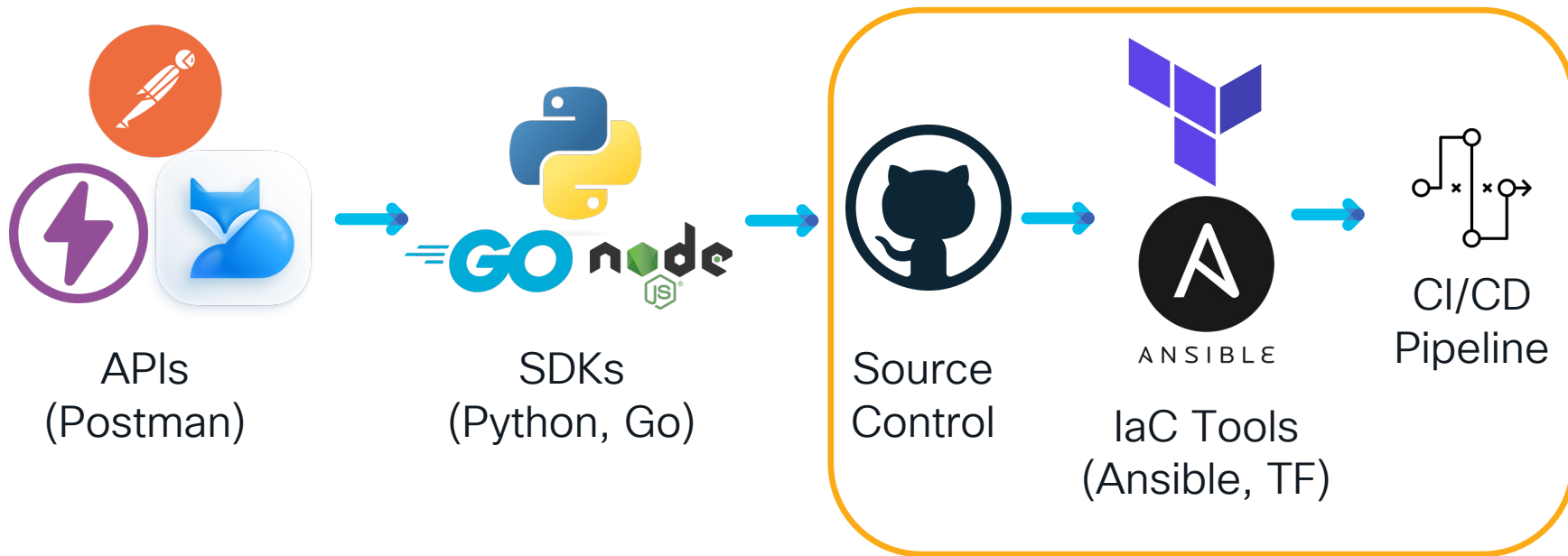
Agenda

- Introduction
- Current state of pipelines
- Abstracting complexity with OTS apps
- Tools of the trade
- Demo
- Conclusion

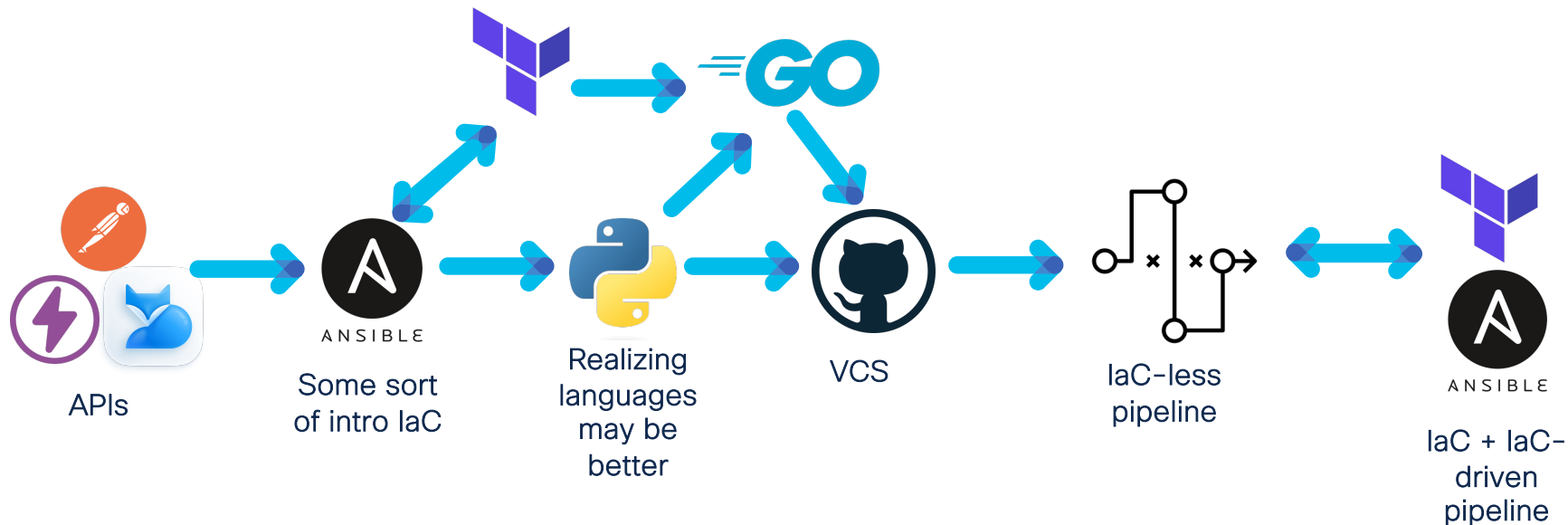
The current state of pipelines



The Progression of Network Automation (so far)



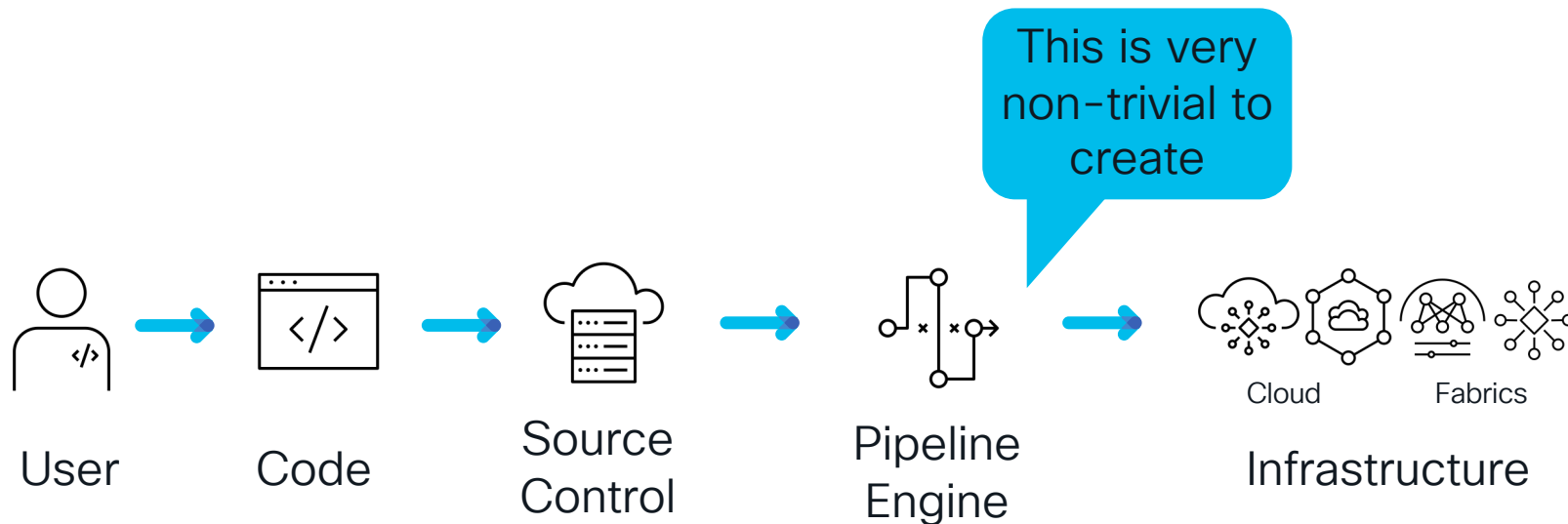
Realistic Network Automation Journey



Why Do I IaC?

- Automate the provisioning and management of the technology stack
- Translate manual tasks into reusable, robust, distributable code
- Rely on practices that have been successfully used for years in software development (version control, automated testing, release tagging, continuous delivery, etc.)
- Benefits: much higher delivery speed; significant reliability boost

The Infrastructure as Code journey



Typical pipeline functions



Linting of source code



Deploy to test/dev/QA environment



Validation of test environment functionality



Deploy to production environment



Validation of production environment

That's a lot of YAML-wrapped-bash

```
stages:
  - validate
  - deploy_to_prod
  - deploy_to_test
  - verify_deploy_to_prod
  - verify_deploy_to_test
  - verify_website_reachability

lint:
  stage: validate
  image: geerlingguy/docker-centos8-ansible:latest
  variables:
  script:
    - ansible-playbook --syntax-check -i inventory/prod.yaml site.yaml
    - ansible-playbook --syntax-check -i inventory/test.yaml site.yaml

deploy_to_prod:
  image: geerlingguy/docker-centos8-ansible:latest
  stage: deploy_to_prod
  script:
    - echo "Deploy to prod env"
    - ansible-playbook -i inventory/prod.yaml site.yaml
  environment:
    name: production
  only:
  - master

deploy_to_test:
  image: geerlingguy/docker-centos8-ansible:latest
  stage: deploy_to_test
  script:
    - echo "Deploy to test env"
    - ansible-playbook -i inventory/test.yaml site.yaml
  environment:
    name: test
  only:
  - test
```

```
verify_test_environment:
  image: ciscotestautomation/pyats:latest-robot
  stage: verify_deploy_to_test
  environment:
    name: test
  script:
    - pwd
    - cd tests
    # Important: need to add our current directory to PYTHONPATH
    - export PYTHONPATH=$PYTHONPATH:$(pwd)
    - robot test.robot

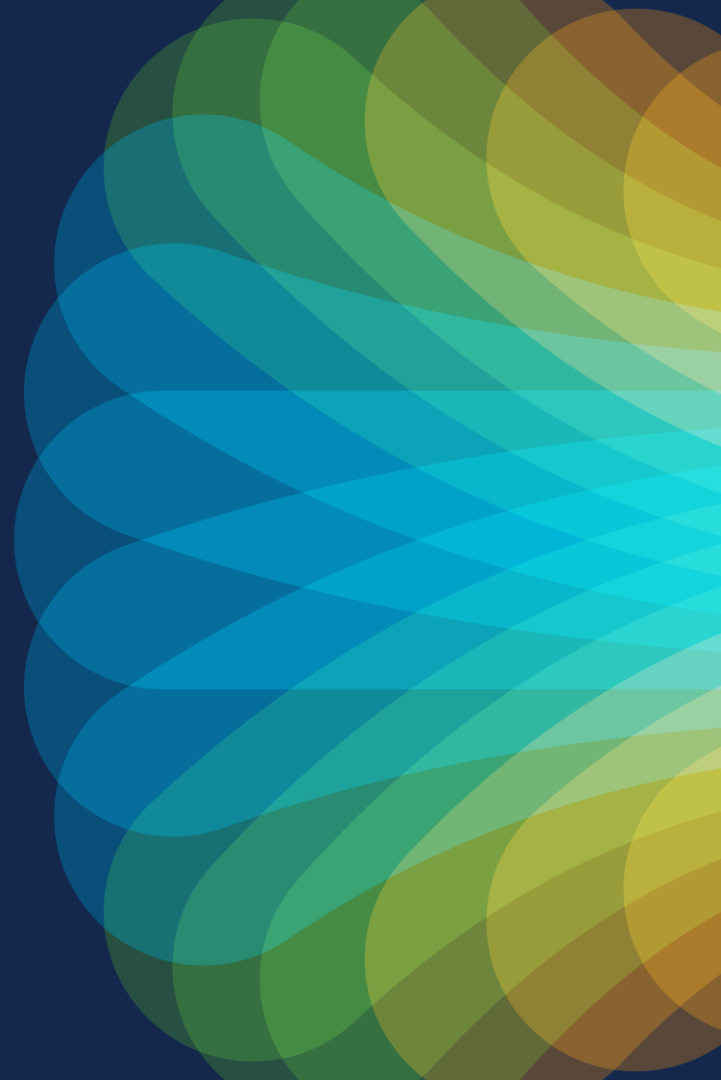
artifacts:
  name: "TEST_${CI_JOB_NAME}_${CI_COMMIT_REF_NAME}"
  when: always
  paths:
    - ./tests/log.html
    - ./tests/report.html
    - ./tests/output.xml
  only:
  - test

verify_prod_environment:
  image: ciscotestautomation/pyats:latest-robot
  stage: verify_deploy_to_prod
  environment:
    name: test
  script:
    - pwd
    - cd tests
    # important: need to add our current directory to PYTHONPATH
    - export PYTHONPATH=$PYTHONPATH:$(pwd)
    - robot prod.robot
```

```
artifacts:
  name: "PROD_${CI_JOB_NAME}_${CI_COMMIT_REF_NAME}"
  when: always
  paths:
    - ./tests/log.html
    - ./tests/report.html
    - ./tests/output.xml
  only:
  - master

internet_sites:
  image: ciscotestautomation/pyats:latest-robot
  stage: verify_website_reachability
  environment:
    name: test
  script:
    - pwd
    - cd tests/websites/
    - make test
  only:
  - master
```

Abstracting complexity with OTS apps

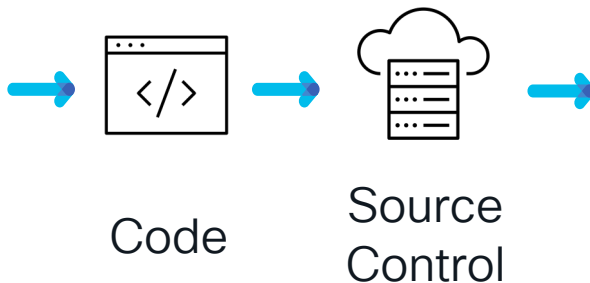


What does this abstraction provide?

- Prebuilt apps perform actions using driver from IaC files
- Common “boiler plate” services (linting, validation) can be handled by app
- Integrates using open methods (webhooks, access tokens) to connect SCM to app and build server
 - BUT -
- Typical OTS platforms require specific tooling/code/etc; loss of flexibility

Our YAML-bash becomes...

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



The image shows a GitHub pull request interface. At the top right is a circular profile picture of a person with a mountain background. The main content area shows a comment from 'qsnryder' 4 minutes ago. The comment text is as follows:

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.

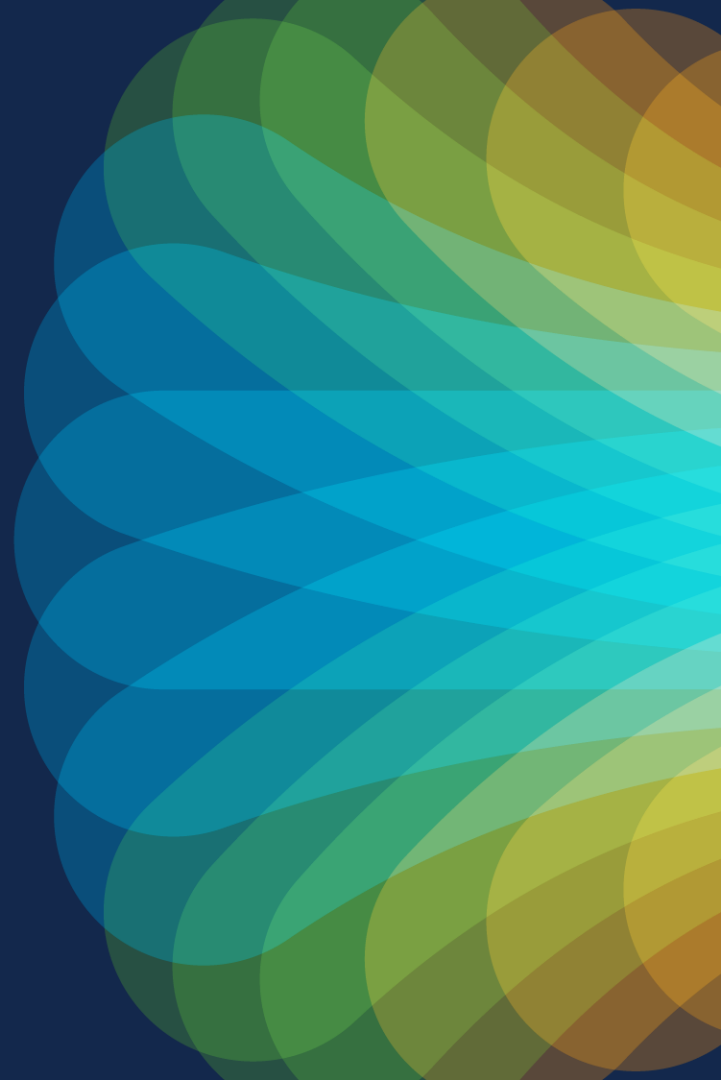
- 🔍 To apply this plan, comment:
 - atlantis apply -d .
- 🗑 To delete this plan click [here](#)
- 🔄 To plan this project again, comment:
 - atlantis plan -d .

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

Below the comment, there are two more sections, each starting with a comment from 'qsnryder' 3 minutes ago. The first section shows the command 'atlantis apply'. The second section shows the output of the apply command:

```
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.
```

Tools of the trade



Terraform

- Open-source infrastructure provisioning tool
- Commercial support from HashiCorp
- Declarative and idempotent
- Immutable infrastructure concept
- Can manage a wide range of systems:
 - VMs, network devices, cloud instances, etc.
- Agentless, single binary file
- Zero server-side dependencies



Atlantis (runatlantis.io)



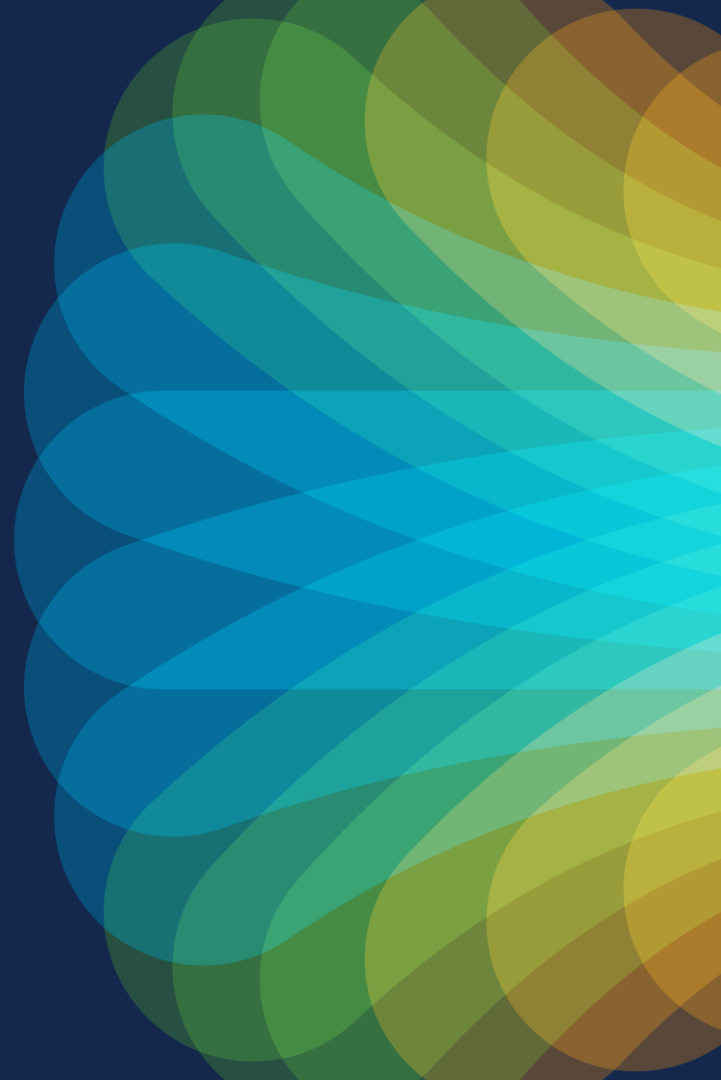
- Open-source tool to automate Terraform workflows using SCM PRs
- Connects to SCM using webhooks, comments results of plan, apply, import into PR
- Runs locally (Go binary or Docker container); no commit of credentials to SCM
- All history logged, full collaboration and visibility amongst teams

- [illegible]



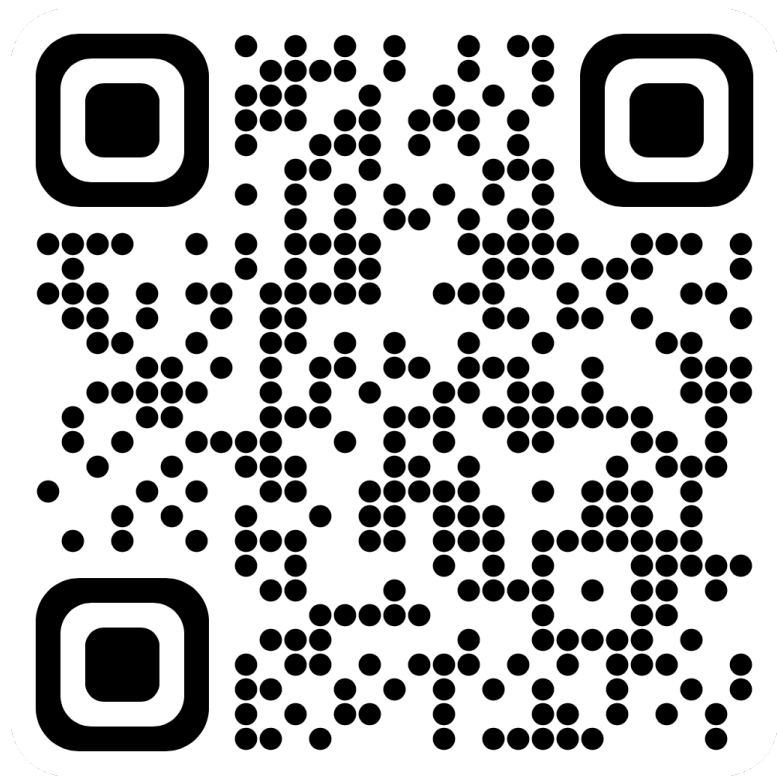
Demo

Bringing it all
together



In conclusion...

- <https://github.com/qsnyder/devnet-3008>
 - QR code is nextdoor
- Contains sample code, scripts, etc to help bootstrap your Atlantis install
- Includes README with cross-links to other software as appropriate





Did you know?

You can have a
one-on-one session with
a technical expert!

Visit Meet the Expert in The HUB
to meet, greet, whiteboard & gain
insights about your unique questions
with the best of the best.



Meet the Expert Opening Hours:

Tuesday	3:00pm – 7:00pm
Wednesday	11:15am – 7:00pm
Thursday	9:30am – 4:00pm
Friday	10:30am – 1:30pm

Session Surveys

We would love to know your feedback on this session!

- Complete a minimum of four session surveys and the overall event surveys to claim a Cisco Live T-Shirt



Continue your education



- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Expert 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

CISCO *Live!*

#CiscoLiveAPJC

The background is a vibrant, abstract graphic featuring a rainbow color palette. On the left, there are horizontal, wavy bands of color transitioning from red at the top to blue at the bottom. On the right, a bright white light source emits a series of sharp, radiating lines in various shades of blue, green, and yellow, creating a sunburst or starburst effect.

cisco *Live!*

Let's go

#CiscoLiveAPJC