Red Hat

Building a Community with Ansible Content Collections

Best practices on developing and maintaining certified Ansible Collection Content.

Carl Buchmann
Senior Solutions Engineer

ARISTA

AnsibleFest

“If you want to go fast, go alone,
if you want to go far, go together”

– African Proverb

“You need to slow down to go faster!”

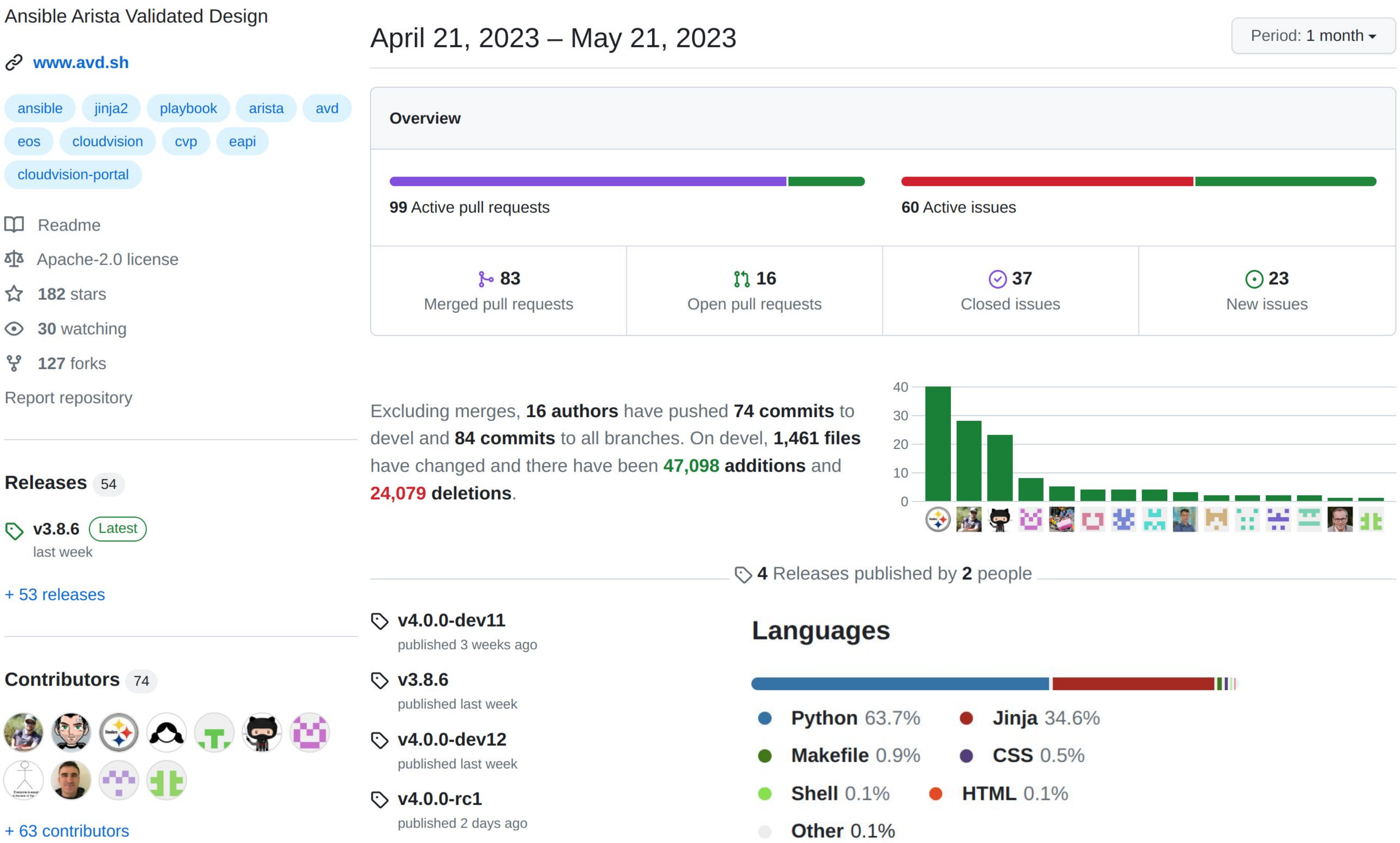
– Viggo Mortensen

- Solutions Engineering Team @ Arista
    - Co-lead/maintainer for Arista AVD
- Previous: Systems Engineer (Networking)
- Using Ansible since ~2018

@carlbuchmann

@carlbuchmann11

arista.avd Ansible Collection



Ansible Arista Validated Design

www.avd.sh

ansible jinja2 playbook arista avd eos cloudvision cvp eapi cloudvision-portal

- Readme
- Apache-2.0 license
- 182 stars
- 30 watching
- 127 forks

Report repository

Releases 54

v3.8.6 Latest
last week

+ 53 releases

Contributors 74

+ 63 contributors

April 21, 2023 – May 21, 2023

Period: 1 month ▾

Overview

99 Active pull requests

60 Active issues

83	16	37	23
Merged pull requests	Open pull requests	Closed issues	New issues

Excluding merges, **16 authors** have pushed **74 commits** to devel and **84 commits** to all branches. On devel, **1,461 files** have changed and there have been **47,098 additions** and **24,079 deletions**.

4 Releases published by **2** people

- **v4.0.0-dev11**
  published 3 weeks ago
- **v3.8.6**
  published last week
- **v4.0.0-dev12**
  published last week
- **v4.0.0-rc1**
  published 2 days ago

Languages

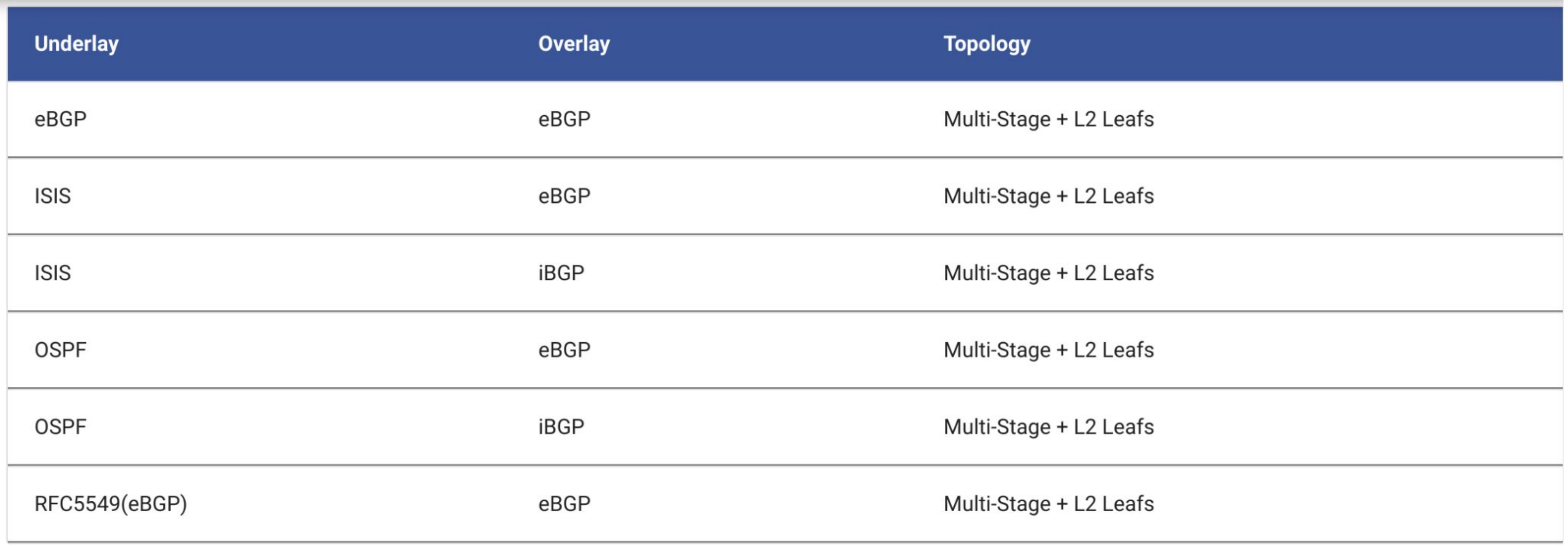
● **Python** 63.7%	● **Jinja** 34.6%
● **Makefile** 0.9%	● **CSS** 0.5%
● **Shell** 0.1%	● **HTML** 0.1%
● **Other** 0.1%	

Ansible Certified
Since November/2020

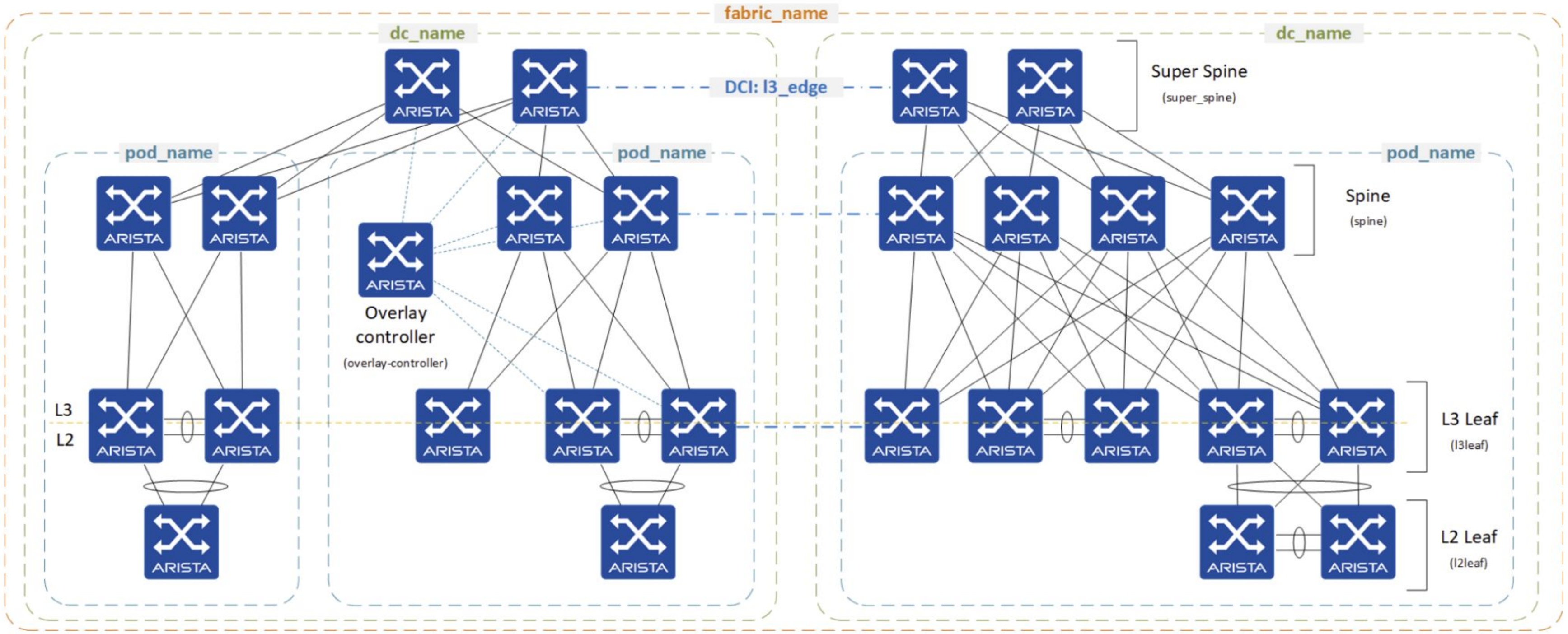
54 Releases

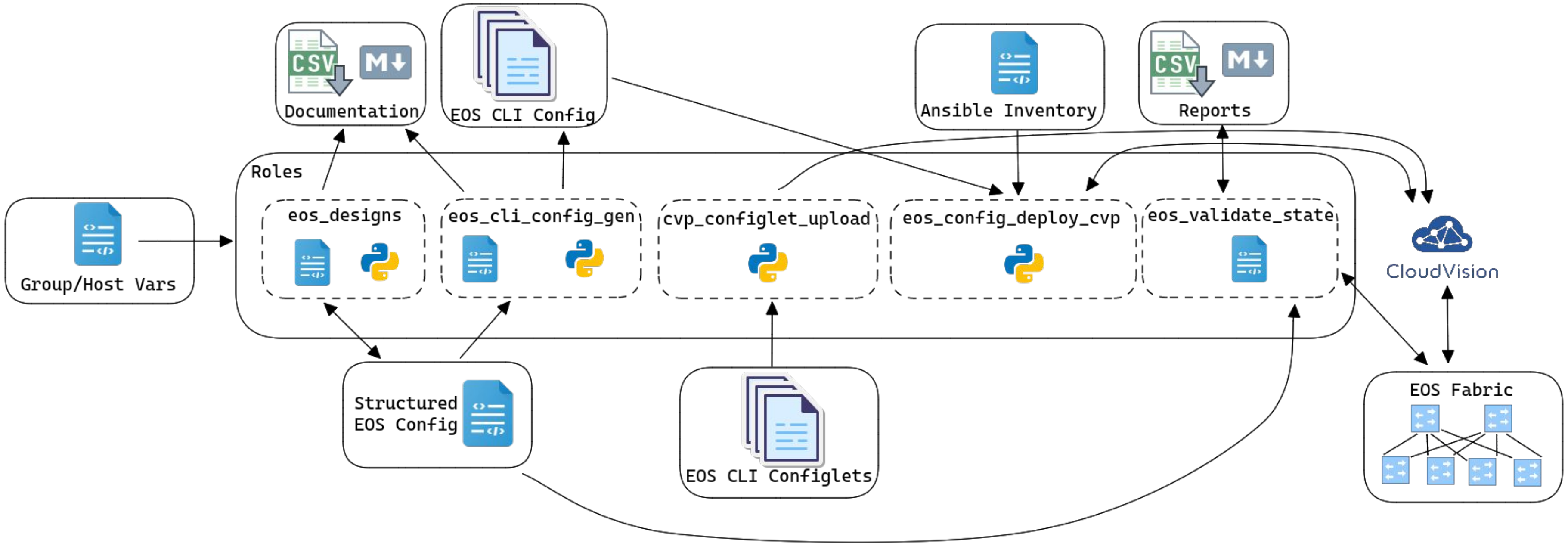
74 Contributors

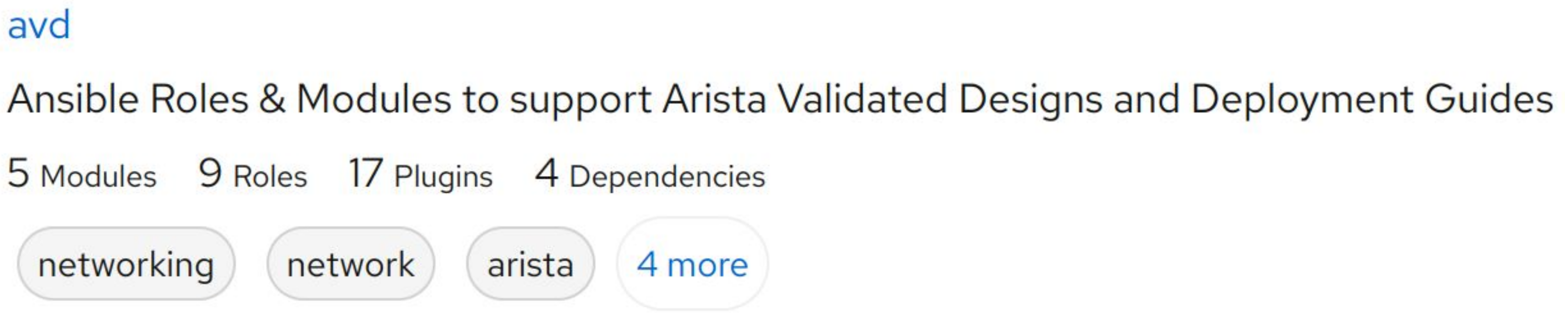
“AVD is a set of **opinionated** design models and templates based on Arista best practices to generate configurations, tests, and documentation for **many network types**”



Underlay	Overlay	Topology
eBGP	eBGP	Multi-Stage + L2 Leafs
ISIS	eBGP	Multi-Stage + L2 Leafs
ISIS	iBGP	Multi-Stage + L2 Leafs
OSPF	eBGP	Multi-Stage + L2 Leafs
OSPF	iBGP	Multi-Stage + L2 Leafs
RFC5549(eBGP)	eBGP	Multi-Stage + L2 Leafs






avd

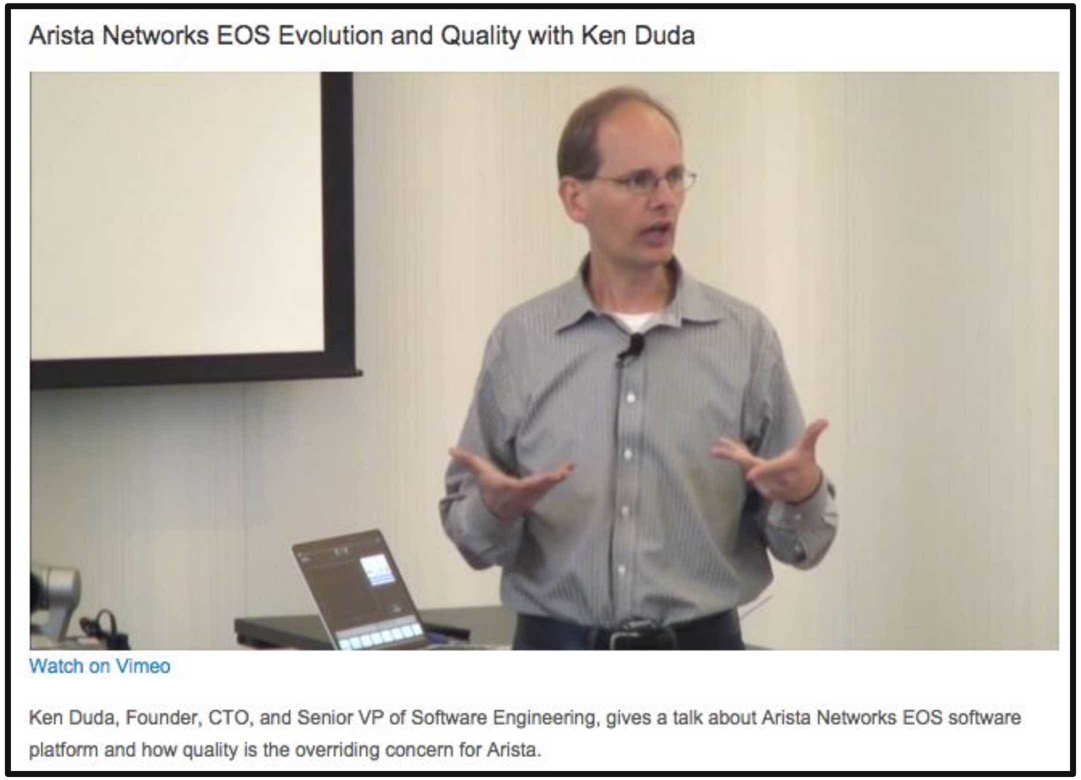
Ansible Roles & Modules to support Arista Validated Designs and Deployment Guides

5 Modules 9 Roles 17 Plugins 4 Dependencies

networking network arista 4 more

Building a community starts with quality

- Culture
- Architecture
- Testing

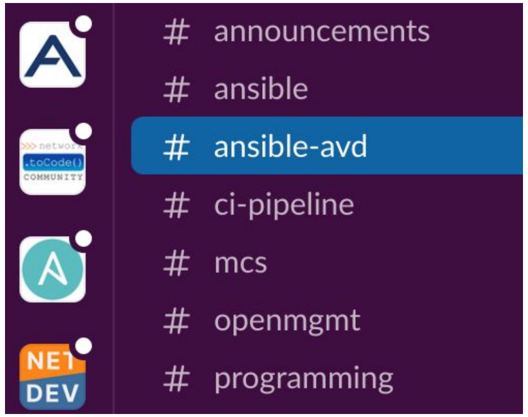


Arista Networks EOS Evolution and Quality with Ken Duda

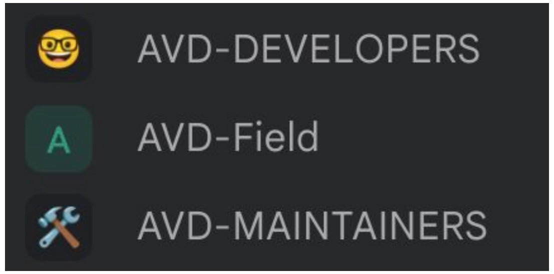
Ken Duda, Founder, CTO, and Senior VP of Software Engineering, gives a talk about Arista Networks EOS software platform and how quality is the overriding concern for Arista.

Arista Networks EOS Evolution and Quality with Ken Duda
Arista's Software Quality Journey with Ken Duda

Team Culture



# announcements
# ansible
# ansible-avd
# ci-pipeline
# mcs
# openmgmt
# programming

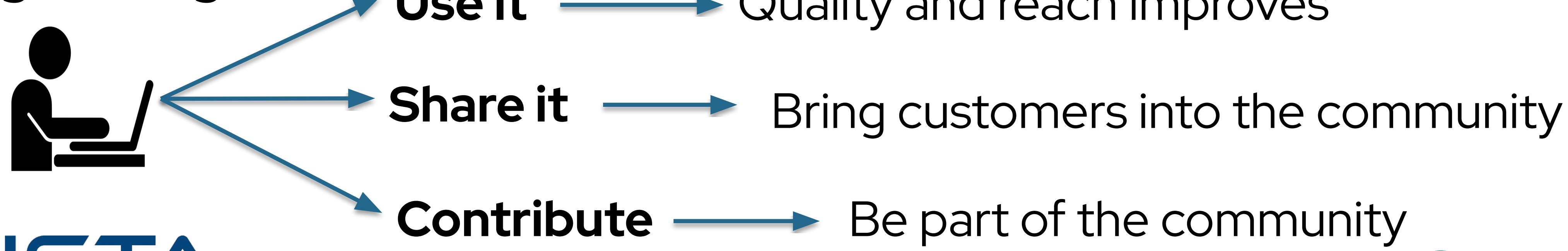


AVD-DEVELOPERS
AVD-Field
AVD-MAINTAINERS

Development Priorities:

- Bug fixes
- Features for customers and projects
- Refactoring and strategic long term initiatives

Arista Customer Engineering

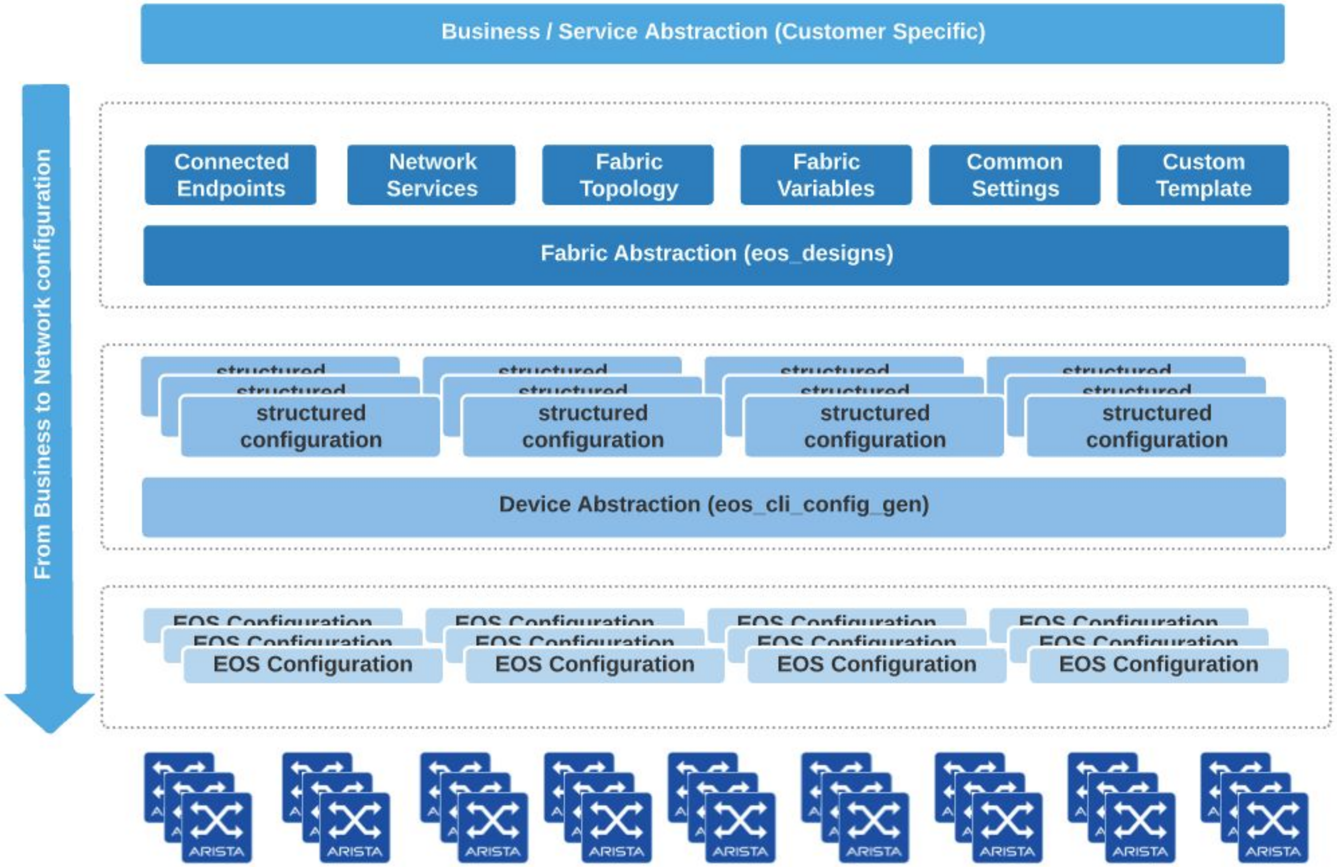


Use it → Quality and reach improves

Share it → Bring customers into the community

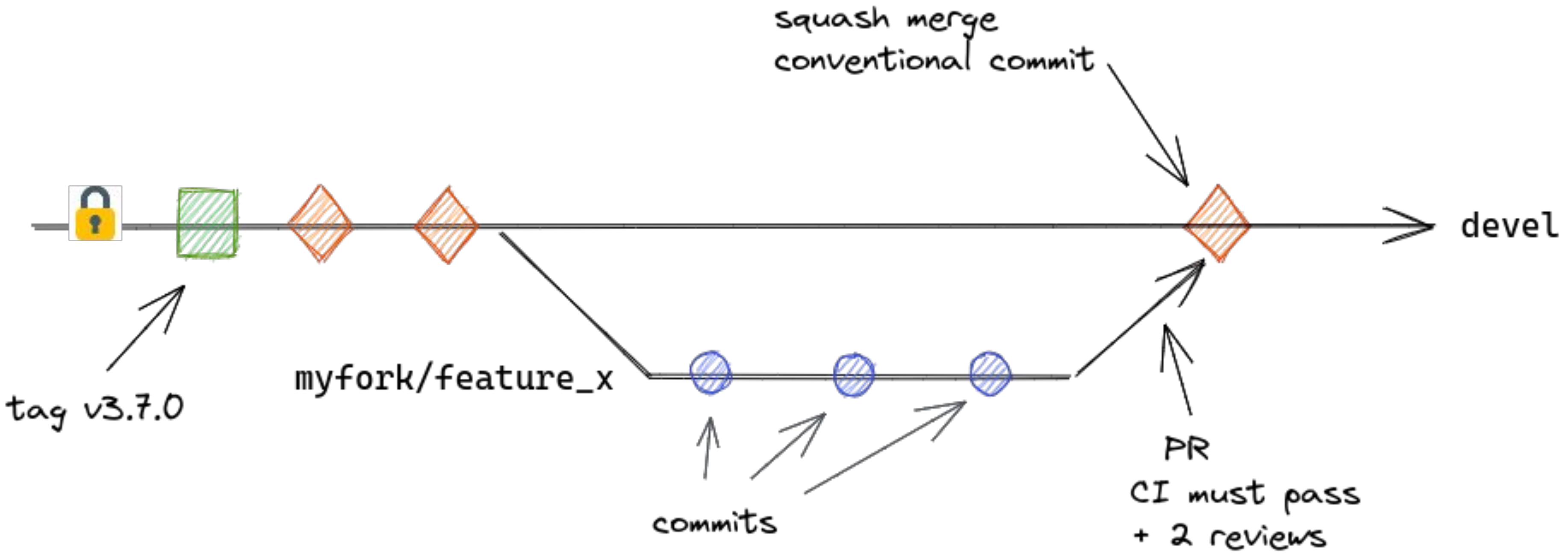
Contribute → Be part of the community

Architecture



- Modular building blocks
- Customer extensible
- Consistent coding style
- Enable customer and developer (typically network eng)
- Documentation as code

Git Branching Strategy



Inspired from: Bits'n'Bites - A stable mainline branching model for Git

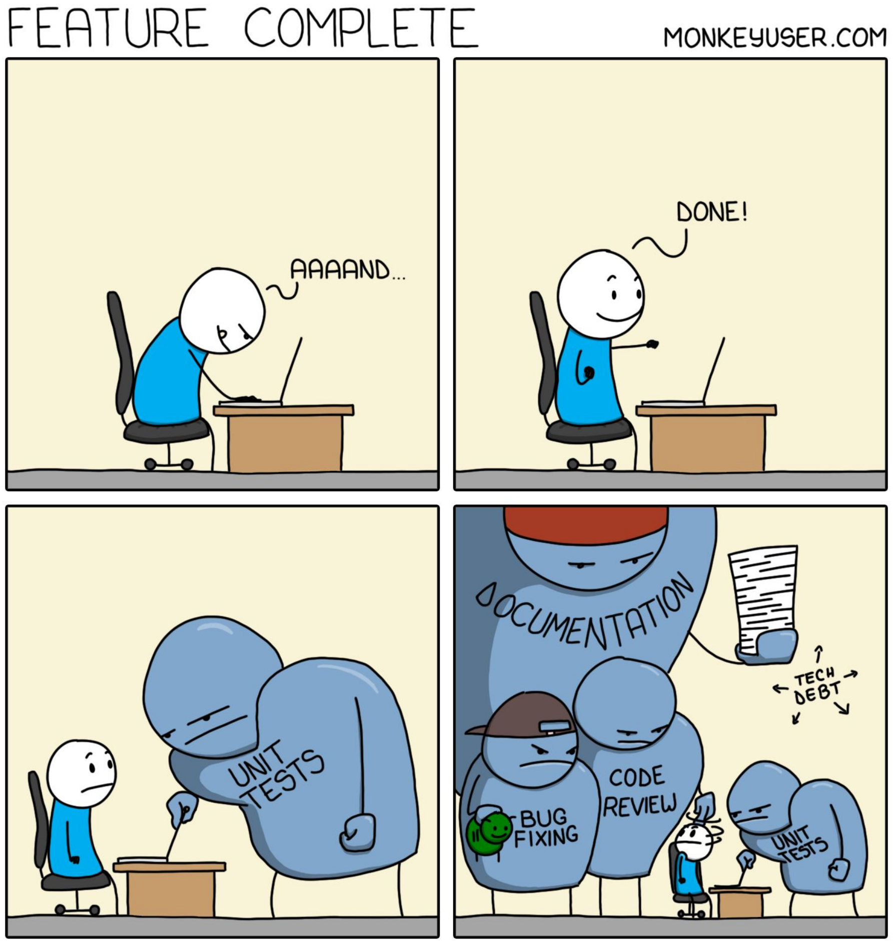
Pull Requests

Should Include:

- Code
- Unit Tests
- Documentation

Tips:

- Keep them small
- Start with a draft PR



Pull Request
Pre-Commit Hooks

- Enforce coding style
  - trailing-whitespace, end-of-file-fixer
  - pylint, yamlint, ansible-lint, markdownlint
  - flake8, Black
  - J2lint
- Schema + documentation

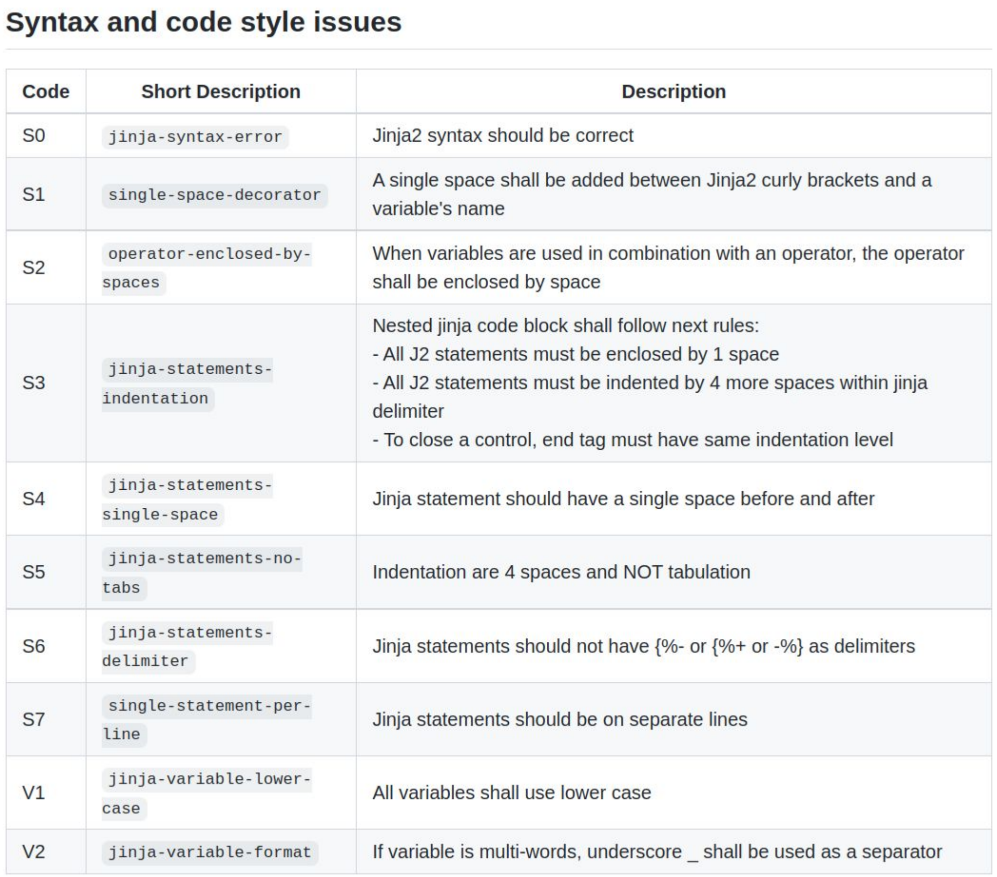
https://github.com/aristanetworks/ansible-avd/blob/devel/.pre-commit-config.yaml

Pro Tip!

Provide guidance in your contribution guide

Arista’s Jinja2-Linter

- Validate syntax according to AVD style guide.
- Capability to run as part of a CI pipeline to enforce j2lint rules.
- `pip install j2lint`

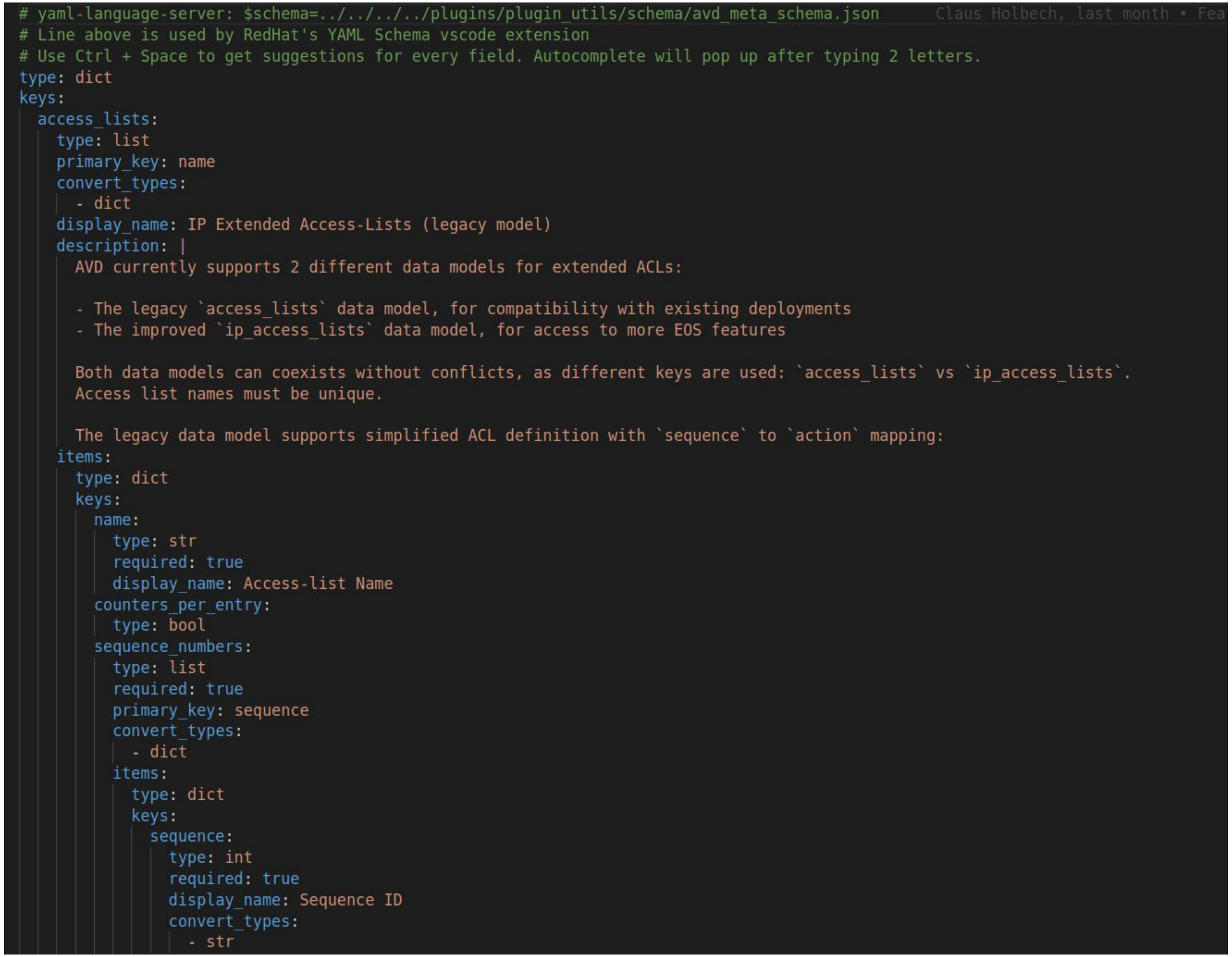


Syntax and code style issues

Code	Short Description	Description
S0	`jinja-syntax-error`	Jinja2 syntax should be correct
S1	`single-space-decorator`	A single space shall be added between Jinja2 curly brackets and a variable's name
S2	`operator-enclosed-by-spaces`	When variables are used in combination with an operator, the operator shall be enclosed by space
S3	`jinja-statements-indentation`	Nested jinja code block shall follow next rules: - All J2 statements must be enclosed by 1 space - All J2 statements must be indented by 4 more spaces within jinja delimiter - To close a control, end tag must have same indentation level
S4	`jinja-statements-single-space`	Jinja statement should have a single space before and after
S5	`jinja-statements-no-tabs`	Indentation are 4 spaces and NOT tabulation
S6	`jinja-statements-delimiter`	Jinja statements should not have {%- or {%+ or -%} as delimiters
S7	`single-statement-per-line`	Jinja statements should be on separate lines
V1	`jinja-variable-lower-case`	All variables shall use lower case
V2	`jinja-variable-format`	If variable is multi-words, underscore _ shall be used as a separator

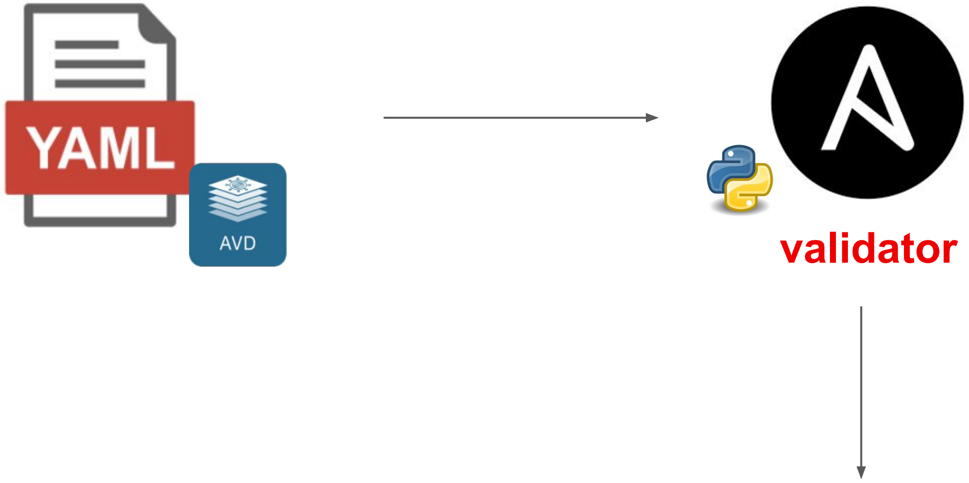
AVD Schema

- **Network Centric**
- Inspired by:
  - JSON Schema
  - Ansible Argument Spec
  - CloudVision Studios
- Provides **unique features** which are not supported or possible with any of the mentioned schema formats
- Support for **automatic type conversion**



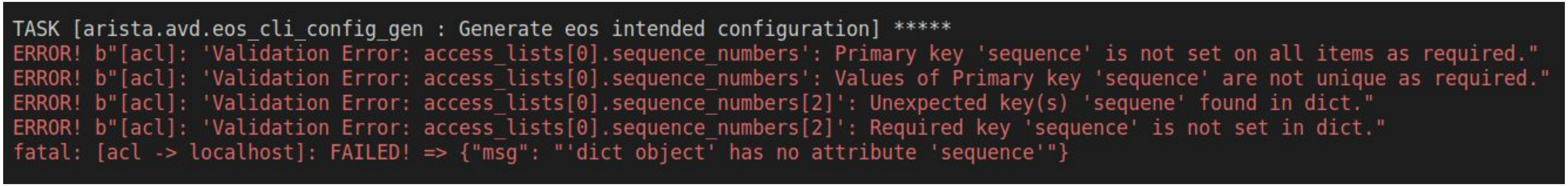
```yaml
# yaml-language-server: $schema=../../../../plugins/plugin_utils/schema/avd_meta_schema.json      Claus Holbech, last month • Fea
# Line above is used by RedHat's YAML Schema vscode extension
# Use Ctrl + Space to get suggestions for every field. Autocomplete will pop up after typing 2 letters.
type: dict
keys:
  access_lists:
    type: list
    primary_key: name
    convert_types:
      - dict
    display_name: IP Extended Access-Lists (legacy model)
    description: |
      AVD currently supports 2 different data models for extended ACLs:

      - The legacy `access_lists` data model, for compatibility with existing deployments
      - The improved `ip_access_lists` data model, for access to more EOS features

      Both data models can coexists without conflicts, as different keys are used: `access_lists` vs `ip_access_lists`.
      Access list names must be unique.

      The legacy data model supports simplified ACL definition with `sequence` to `action` mapping:
    items:
      type: dict
      keys:
        name:
          type: str
          required: true
          display_name: Access-list Name
        counters_per_entry:
          type: bool
        sequence_numbers:
          type: list
          required: true
          primary_key: sequence
          convert_types:
            - dict
          items:
            type: dict
            keys:
              sequence:
                type: int
                required: true
                display_name: Sequence ID
                convert_types:
                  - str
```

AVD Schema Validator



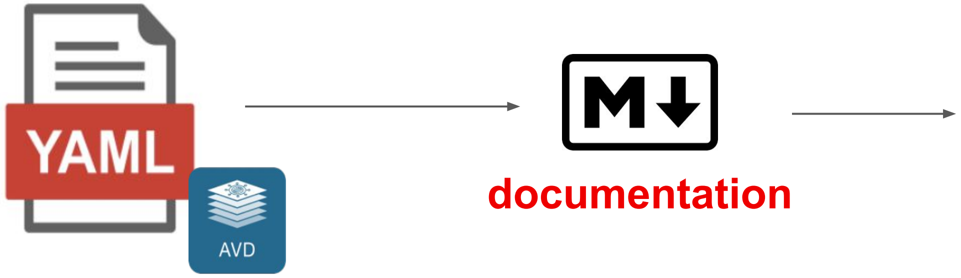
Validates:

- Primary key is set
- Primary key is unique*
- Incorrect key
- Required keys
- Data type
- Etc.
- Ability to develop, and extend as required

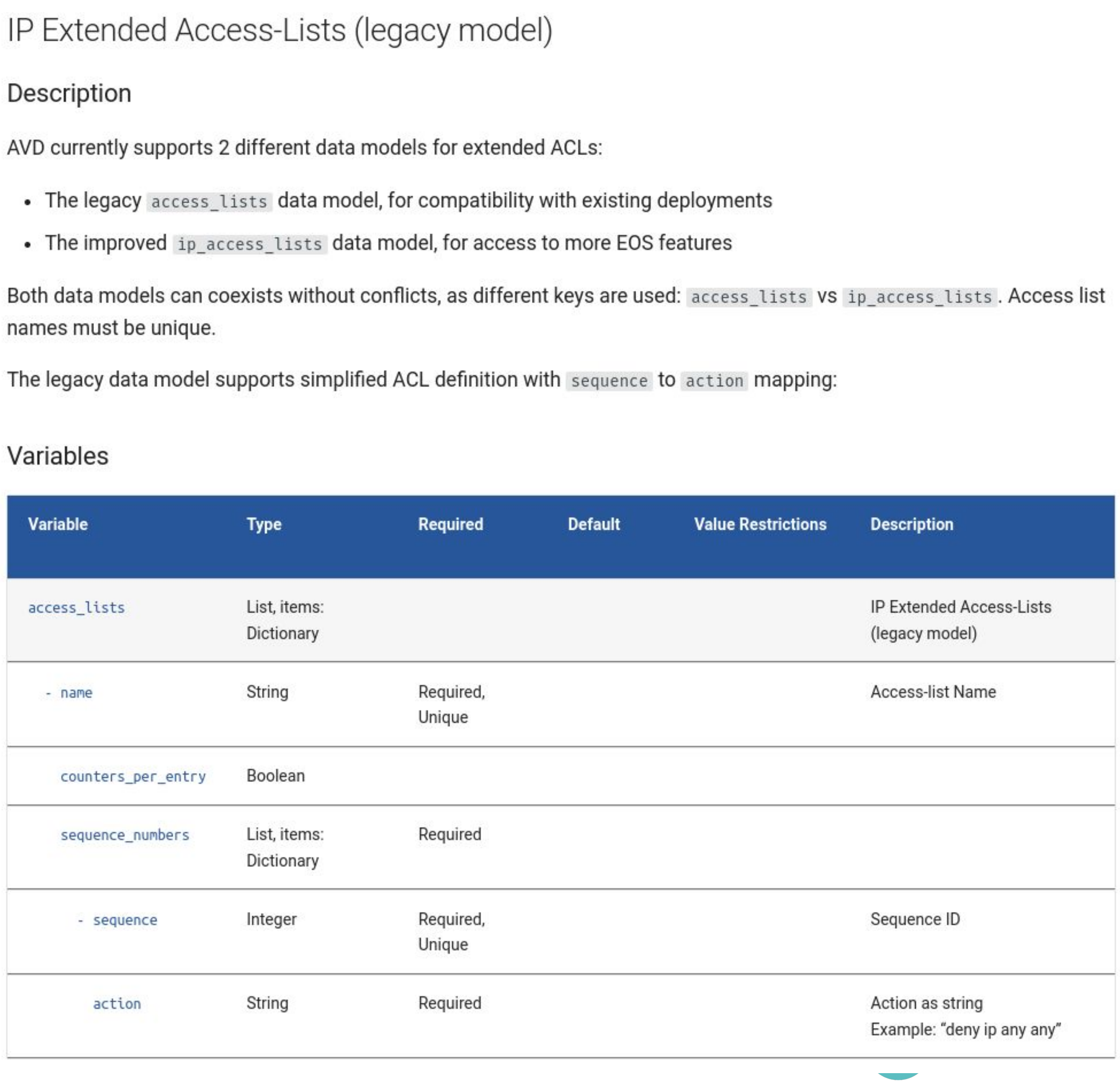


```
TASK [arista.avd.eos_cli_config_gen : Generate eos intended configuration] *****
ERROR! b"[acl]: 'Validation Error: access_lists[0].sequence_numbers': Primary key 'sequence' is not set on all items as required."
ERROR! b"[acl]: 'Validation Error: access_lists[0].sequence_numbers': Values of Primary key 'sequence' are not unique as required."
ERROR! b"[acl]: 'Validation Error: access_lists[0].sequence_numbers[2]': Unexpected key(s) 'sequene' found in dict."
ERROR! b"[acl]: 'Validation Error: access_lists[0].sequence_numbers[2]': Required key 'sequence' is not set in dict."
fatal: [acl -> localhost]: FAILED! => {"msg": "'dict object' has no attribute 'sequence'"}
```

AVD Schema Documentation



Documentation is **automatically generated** from schema, ensuring consistency and **improving user documentation**



IP Extended Access-Lists (legacy model)

Description

AVD currently supports 2 different data models for extended ACLs:

- The legacy `access_lists` data model, for compatibility with existing deployments
- The improved `ip_access_lists` data model, for access to more EOS features

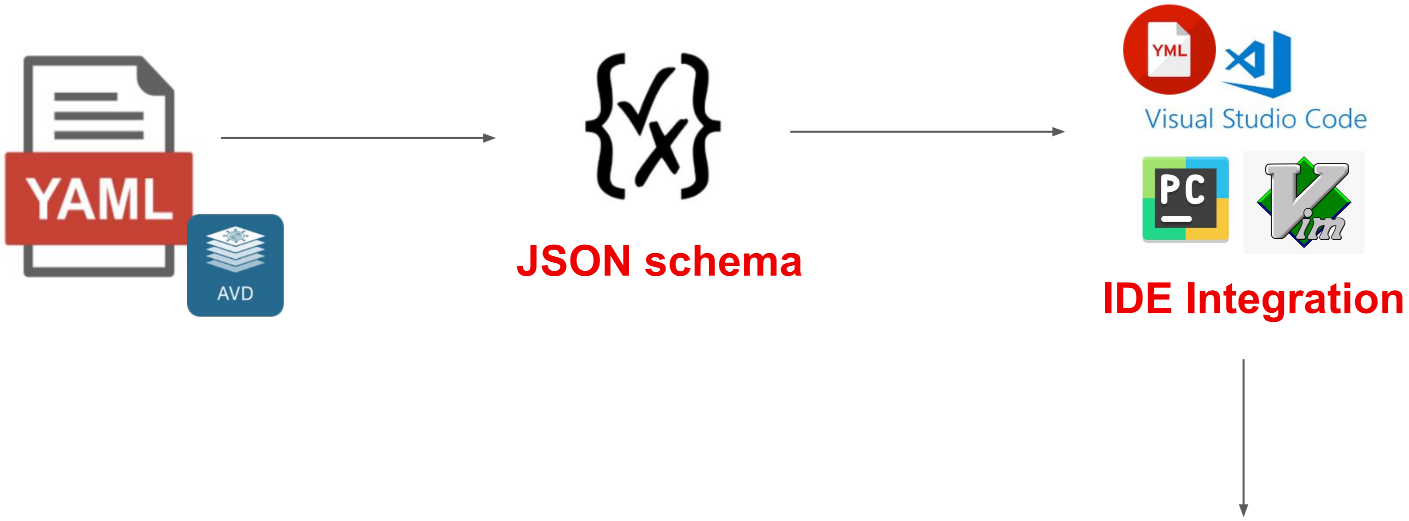
Both data models can coexists without conflicts, as different keys are used: `access_lists` vs `ip_access_lists`. Access list names must be unique.

The legacy data model supports simplified ACL definition with `sequence` to `action` mapping:

Variables

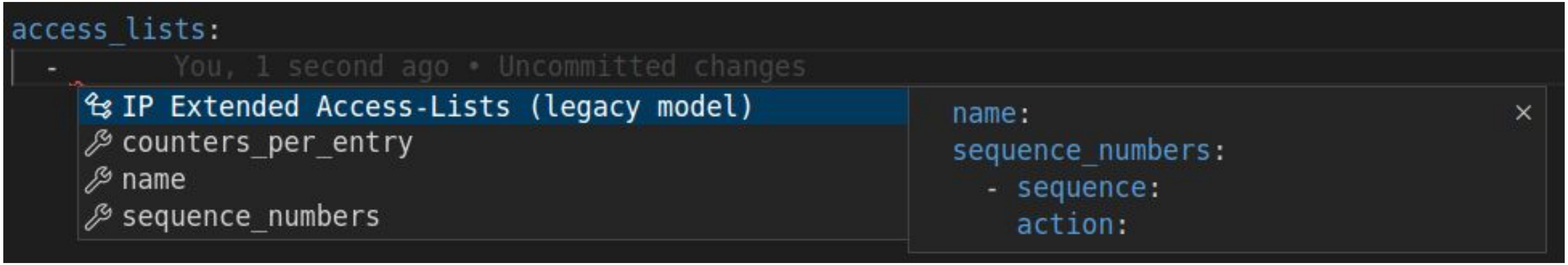
Variable	Type	Required	Default	Value Restrictions	Description
`access_lists`	List, items: Dictionary				IP Extended Access-Lists (legacy model)
`- name`	String	Required, Unique			Access-list Name
`counters_per_entry`	Boolean				
`sequence_numbers`	List, items: Dictionary	Required			
`- sequence`	Integer	Required, Unique			Sequence ID
`action`	String	Required			Action as string Example: "deny ip any any"

AVD Schema IDE Integration



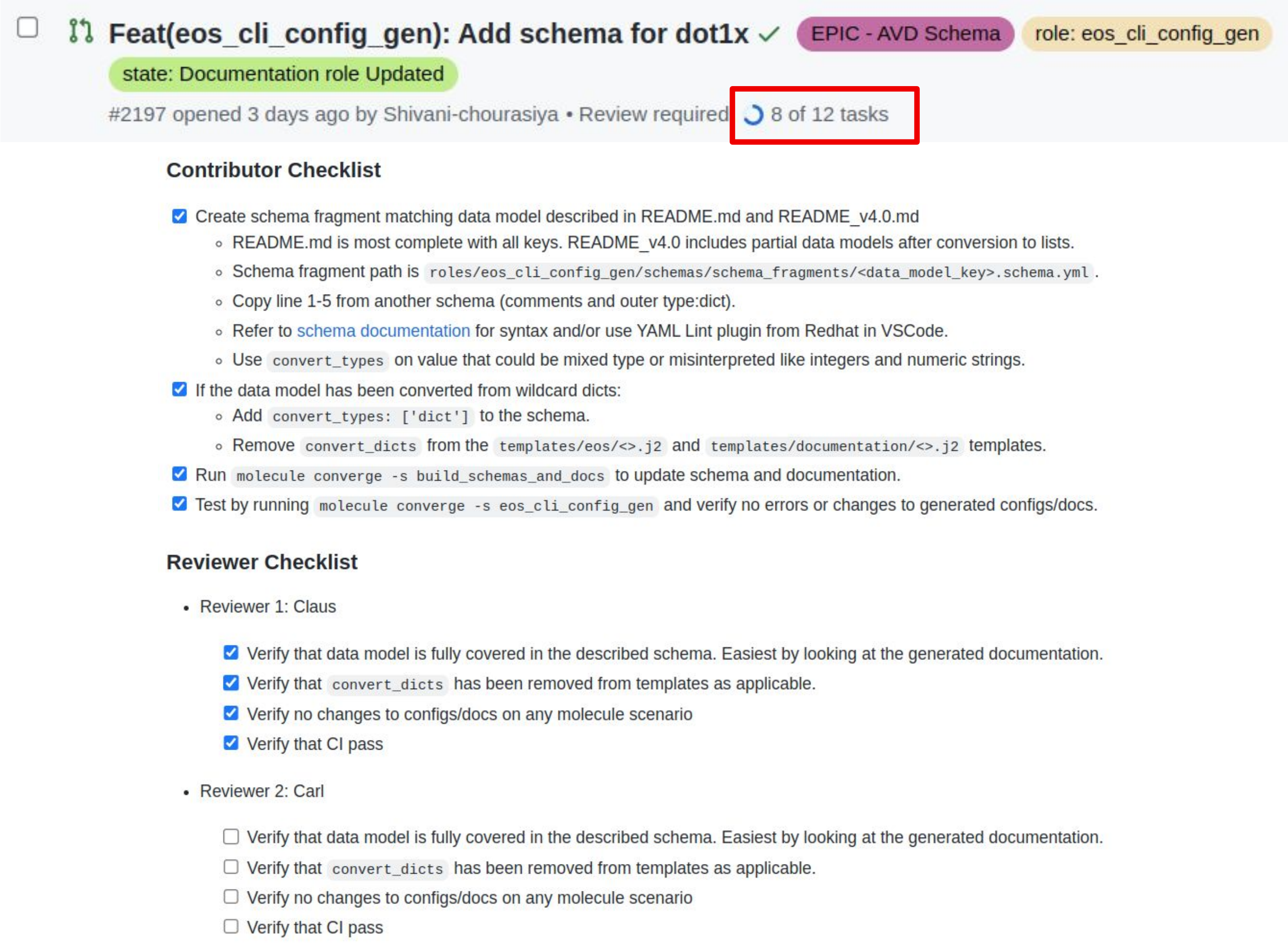
JSON schema

IDE Integration



```
access_lists:
  -          You, 1 second ago • Uncommitted changes
     IP Extended Access-Lists (legacy model)    name:
     counters_per_entry                         sequence_numbers:
     name                                         - sequence:
     sequence_numbers                               action:
```

PR Checklists



□ Feat(eos_cli_config_gen): Add schema for dot1x ✓ EPIC - AVD Schema role: eos_cli_config_gen state: Documentation role Updated

#2197 opened 3 days ago by Shivani-chourasiya • Review required 8 of 12 tasks

Contributor Checklist

- ☑ Create schema fragment matching data model described in README.md and README_v4.0.md
  - README.md is most complete with all keys. README_v4.0 includes partial data models after conversion to lists.
  - Schema fragment path is `roles/eos_cli_config_gen/schemas/schema_fragments/<data_model_key>.schema.yml`.
  - Copy line 1-5 from another schema (comments and outer type:dict).
  - Refer to schema documentation for syntax and/or use YAML Lint plugin from Redhat in VSCode.
  - Use `convert_types` on value that could be mixed type or misinterpreted like integers and numeric strings.
- ☑ If the data model has been converted from wildcard dicts:
  - Add `convert_types: ['dict']` to the schema.
  - Remove `convert_dicts` from the `templates/eos/<>.j2` and `templates/documentation/<>.j2` templates.
- ☑ Run `molecule converge -s build_schemas_and_docs` to update schema and documentation.
- ☑ Test by running `molecule converge -s eos_cli_config_gen` and verify no errors or changes to generated configs/docs.

Reviewer Checklist

- Reviewer 1: Claus
  - ☑ Verify that data model is fully covered in the described schema. Easiest by looking at the generated documentation.
  - ☑ Verify that `convert_dicts` has been removed from templates as applicable.
  - ☑ Verify no changes to configs/docs on any molecule scenario
  - ☑ Verify that CI pass
- Reviewer 2: Carl
  - □ Verify that data model is fully covered in the described schema. Easiest by looking at the generated documentation.
  - □ Verify that `convert_dicts` has been removed from templates as applicable.
  - □ Verify no changes to configs/docs on any molecule scenario
  - □ Verify that CI pass

Pro Tip!

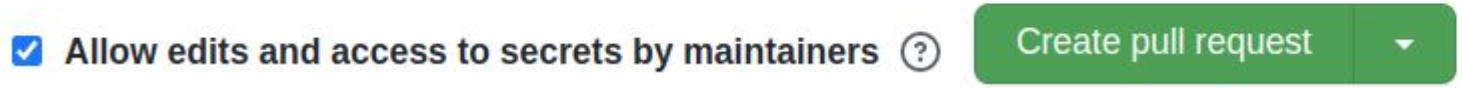
GitHub Repositories lets you store multiple pull request templates in the “PULL_REQUEST_TEMPLATE” directory.

GitHub Pull Request Templates

AnsibleFest

Pull Request: Peer Review

- Each PR requires two reviews
- Review of code and documentation
- Review and test functionality
- Collaborate on PR
- **CI must pass**



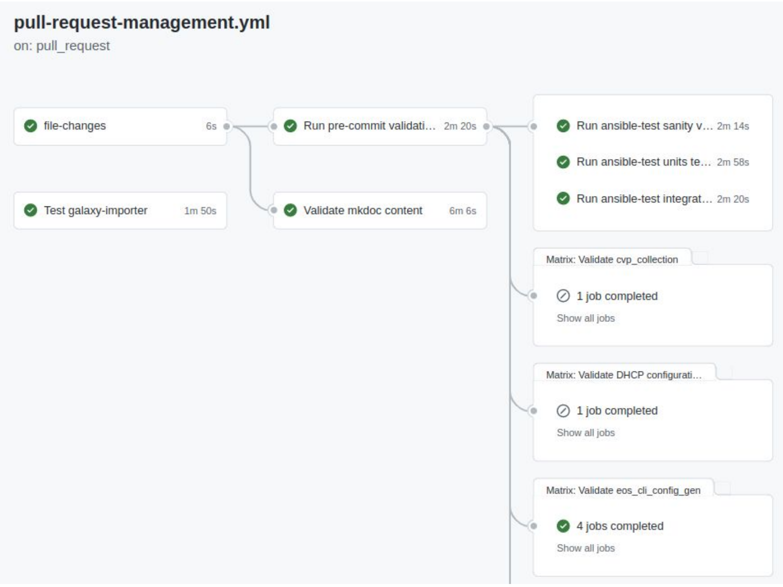
☑ Allow edits and access to secrets by maintainers ?

Pro Tip!

Recommend contributors to give maintainer access when submitting PR.

CI Pipeline

- **Leverage Ansible Tooling**
  - galaxy-importer
  - Ansible-test
    (sanity, units, integration)
  - molecule
- Run pre-commit validation
- Validate mkdoc content



pull-request-management.yml
on: pull_request

✓ file-changes 6s
✓ Test galaxy-importer 1m 50s
✓ Run pre-commit validati… 2m 20s
✓ Validate mkdoc content 6m 6s
✓ Run ansible-test sanity v… 2m 14s
✓ Run ansible-test units te… 2m 58s
✓ Run ansible-test integrat… 2m 20s

Matrix: Validate cvp_collection
⊘ 1 job completed
Show all jobs

Matrix: Validate DHCP configurati…
⊘ 1 job completed
Show all jobs

Matrix: Validate eos_cli_config_gen
✓ 4 jobs completed
Show all jobs

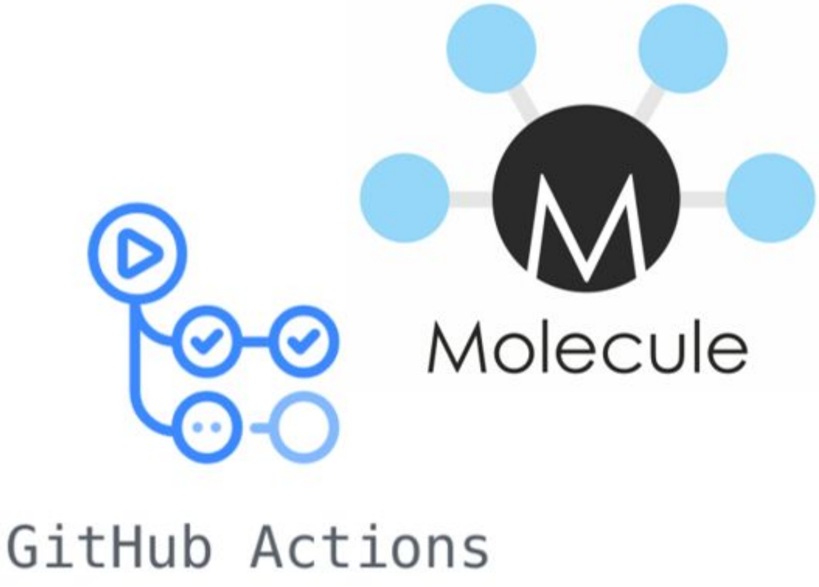
https://github.com/aristanetworks/ansible-avd/blob/devel/.github/workflows/pull-request-management.yml

Pro Tip!

Leverage a Makefile to allow contributors to easily run tests locally.

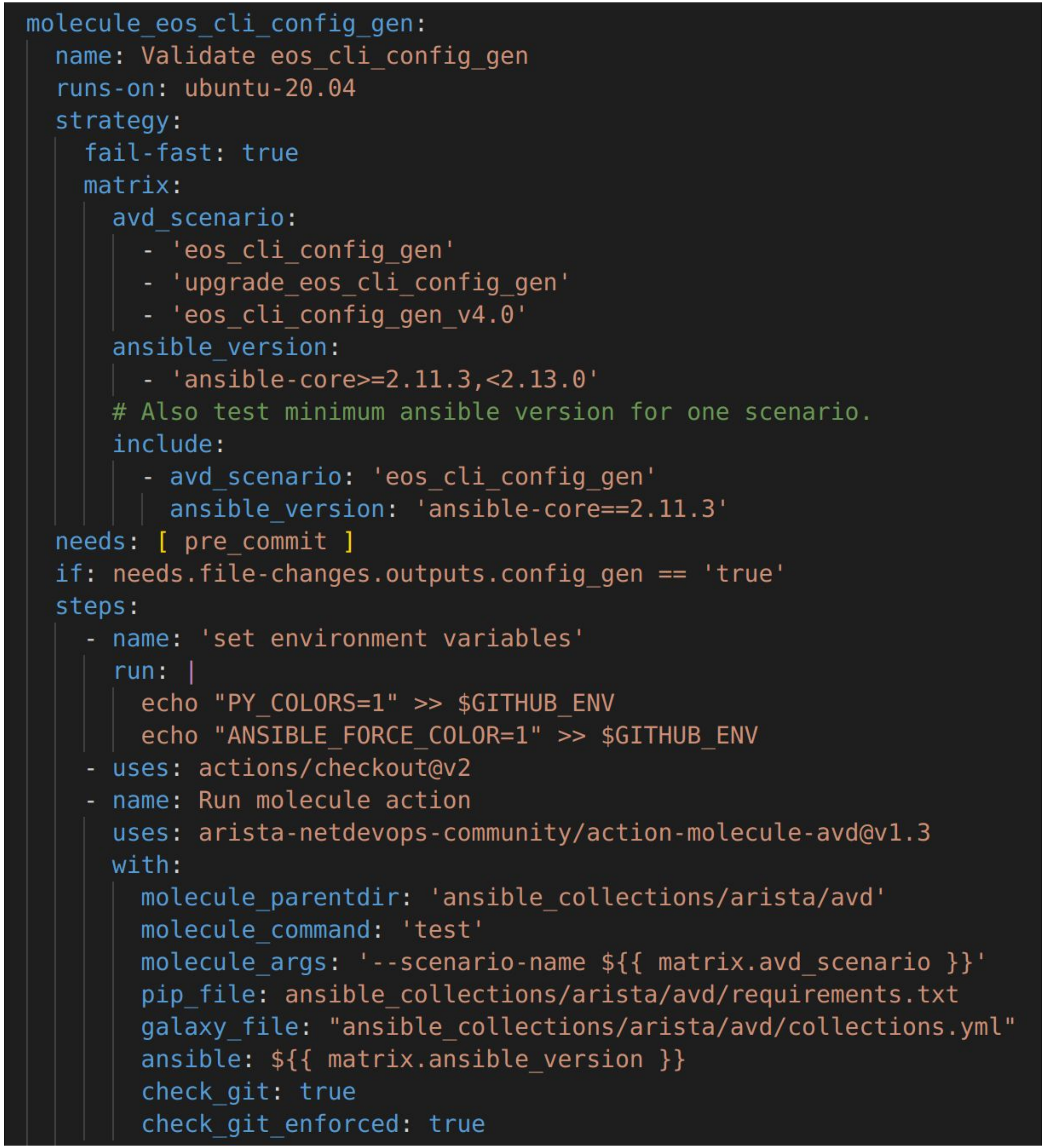
GitHub Action Molecule

- Simplifies unit test creation, enabling your contributors and reach
- Allows to run Molecule using Ansible collection structure
- Supports Git status after molecule execution to track unexpected changes



https://github.com/arista-netdevops-community/action-molecule-avd

Testing: CI Pipeline



```yaml
molecule_eos_cli_config_gen:
  name: Validate eos_cli_config_gen
  runs-on: ubuntu-20.04
  strategy:
    fail-fast: true
    matrix:
      avd_scenario:
        - 'eos_cli_config_gen'
        - 'upgrade_eos_cli_config_gen'
        - 'eos_cli_config_gen_v4.0'
      ansible_version:
        - 'ansible-core>=2.11.3,<2.13.0'
      # Also test minimum ansible version for one scenario.
      include:
        - avd_scenario: 'eos_cli_config_gen'
          ansible_version: 'ansible-core==2.11.3'
  needs: [ pre_commit ]
  if: needs.file-changes.outputs.config_gen == 'true'
  steps:
    - name: 'set environment variables'
      run: |
        echo "PY_COLORS=1" >> $GITHUB_ENV
        echo "ANSIBLE_FORCE_COLOR=1" >> $GITHUB_ENV
    - uses: actions/checkout@v2
    - name: Run molecule action
      uses: arista-netdevops-community/action-molecule-avd@v1.3
      with:
        molecule_parentdir: 'ansible_collections/arista/avd'
        molecule_command: 'test'
        molecule_args: '--scenario-name ${{ matrix.avd_scenario }}'
        pip_file: ansible_collections/arista/avd/requirements.txt
        galaxy_file: "ansible_collections/arista/avd/collections.yml"
        ansible: ${{ matrix.ansible_version }}
        check_git: true
        check_git_enforced: true
```

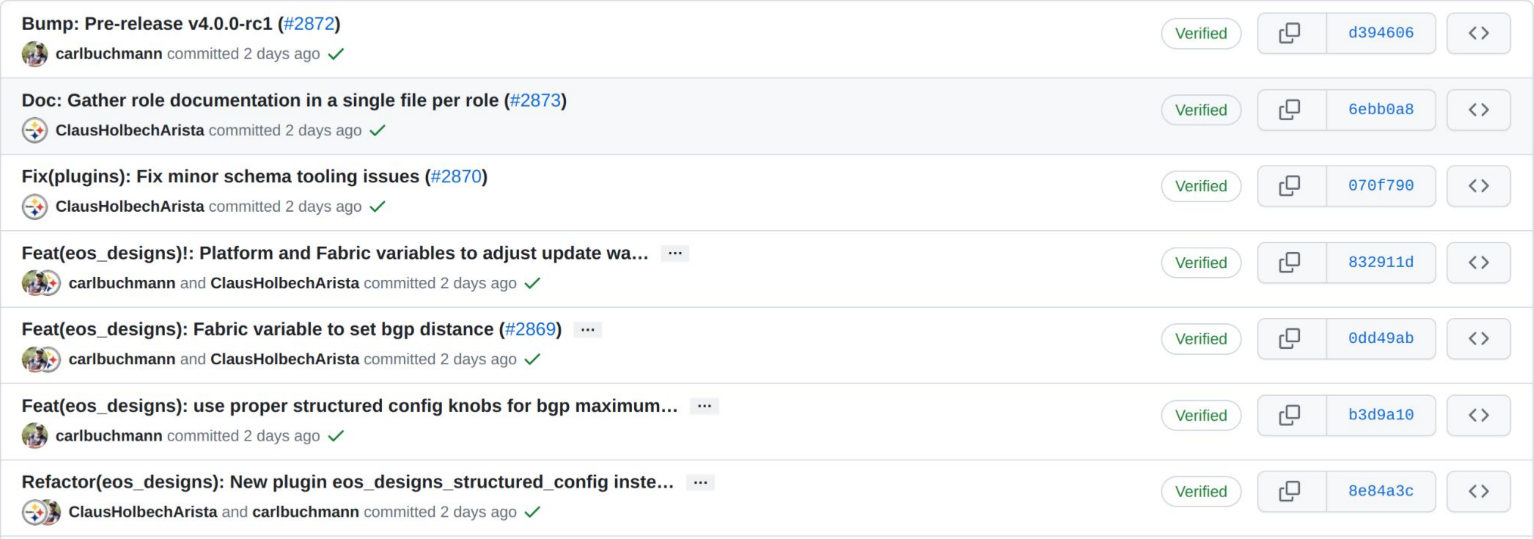
Merge Pull Request

Squash merge

- Condenses the git history into one commit

Conventional commit

- A specification for adding human and machine readable meaning to commit messages



Bump: Pre-release v4.0.0-rc1 (#2872)
carlbuchmann committed 2 days ago ✓ — Verified — d394606

Doc: Gather role documentation in a single file per role (#2873)
ClausHolbechArista committed 2 days ago ✓ — Verified — 6ebb0a8

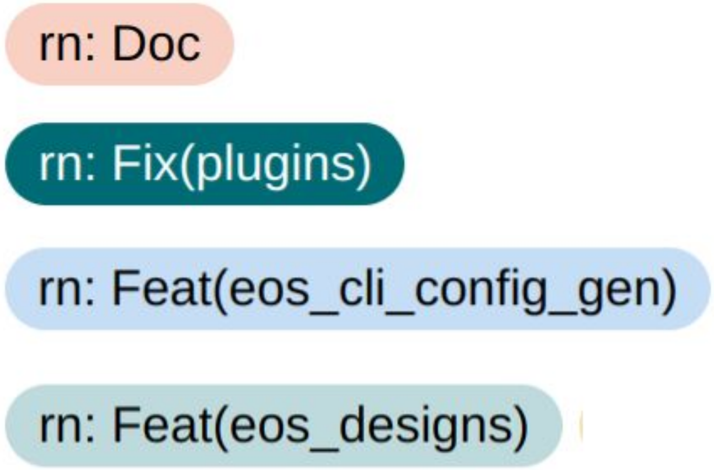
Fix(plugins): Fix minor schema tooling issues (#2870)
ClausHolbechArista committed 2 days ago ✓ — Verified — 070f790

Feat(eos_designs)!: Platform and Fabric variables to adjust update wa... ···
carlbuchmann and ClausHolbechArista committed 2 days ago ✓ — Verified — 832911d

Feat(eos_designs): Fabric variable to set bgp distance (#2869) ···
carlbuchmann and ClausHolbechArista committed 2 days ago ✓ — Verified — 0dd49ab

Feat(eos_designs): use proper structured config knobs for bgp maximum... ···
carlbuchmann committed 2 days ago ✓ — Verified — b3d9a10

Refactor(eos_designs): New plugin eos_designs_structured_config inste... ···
ClausHolbechArista and carlbuchmann committed 2 days ago ✓ — Verified — 8e84a3c



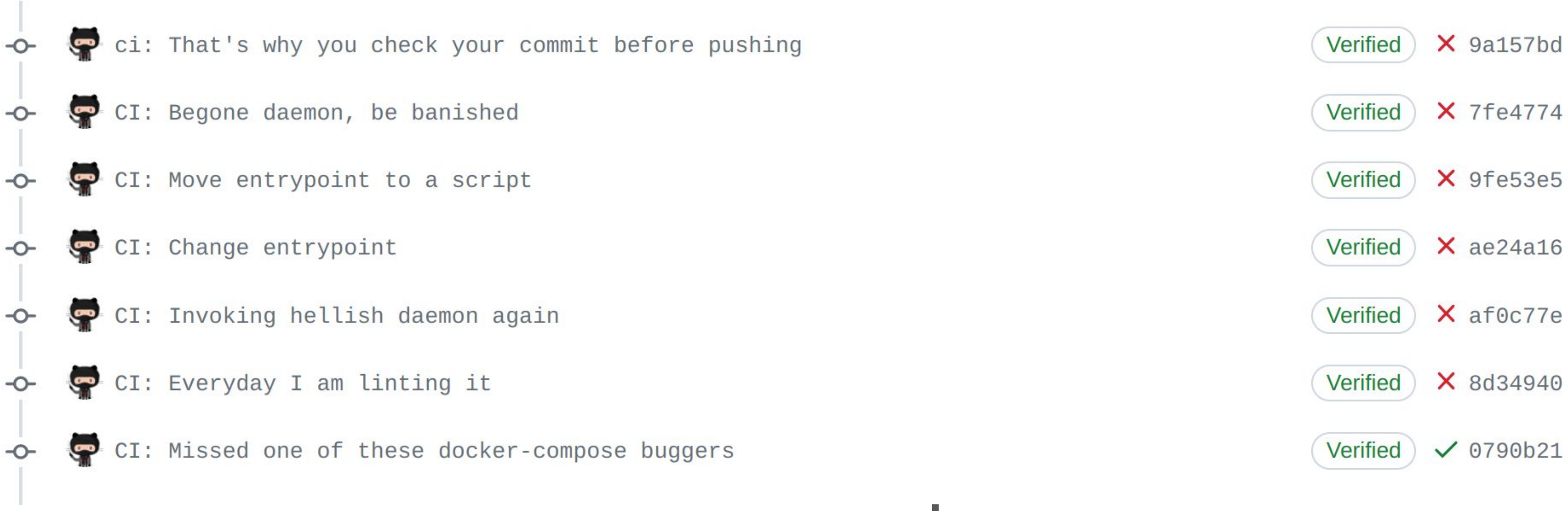
rn: Doc

rn: Fix(plugins)

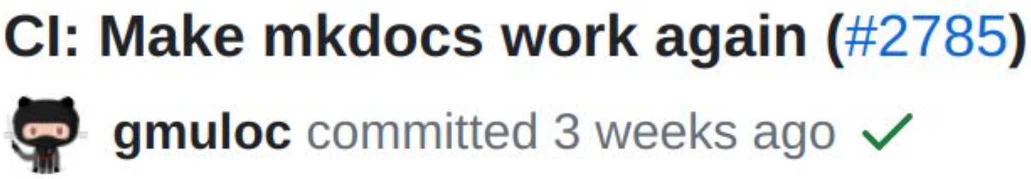
rn: Feat(eos_cli_config_gen)

rn: Feat(eos_designs)

Squash Merge in Practice



```
ci: That's why you check your commit before pushing      Verified  ✗ 9a157bd
CI: Begone daemon, be banished                           Verified  ✗ 7fe4774
CI: Move entrypoint to a script                          Verified  ✗ 9fe53e5
CI: Change entrypoint                                    Verified  ✗ ae24a16
CI: Invoking hellish daemon again                        Verified  ✗ af0c77e
CI: Everyday I am linting it                             Verified  ✗ 8d34940
CI: Missed one of these docker-compose buggers           Verified  ✓ 0790b21
```



CI: Make mkdocs work again (#2785)

gmuloc committed 3 weeks ago ✓

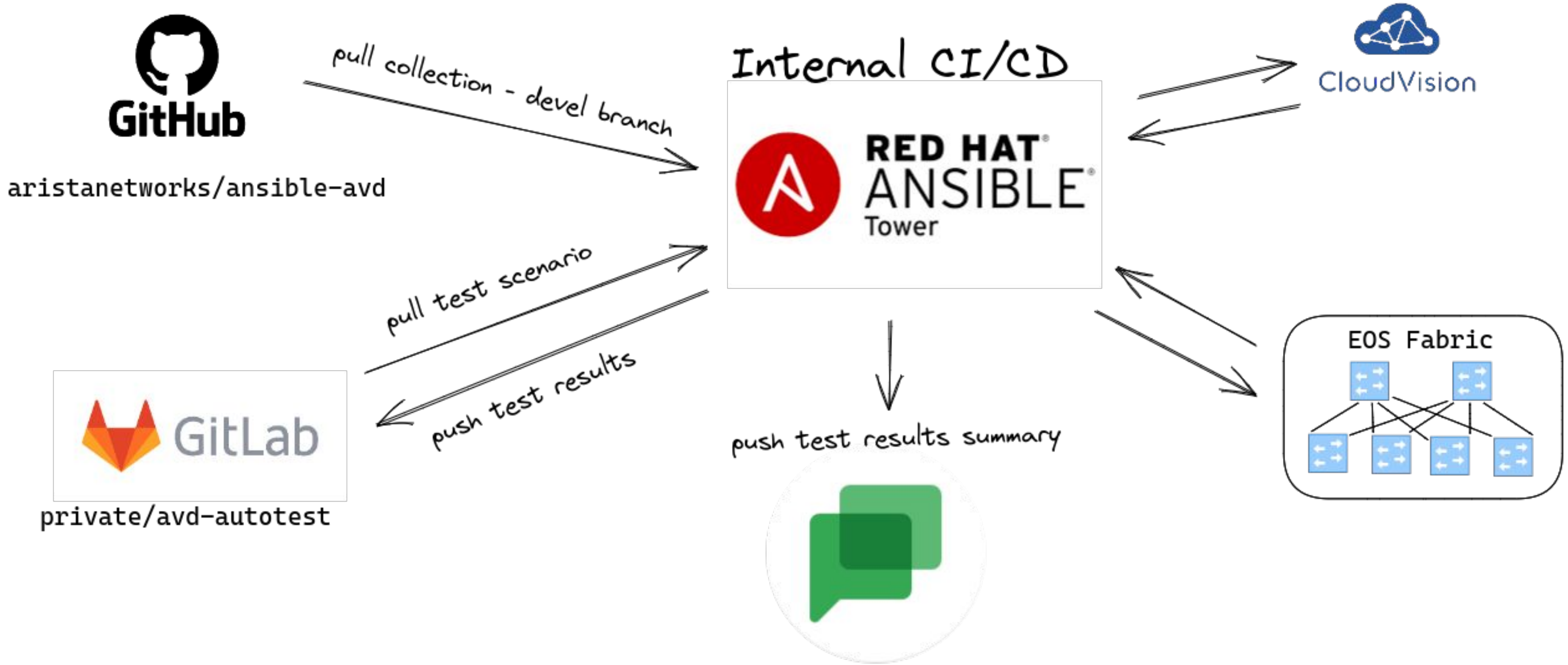
Project Documentation

- Hosted on Read the Docs
  - Leveraging Material theme for MKDocs.

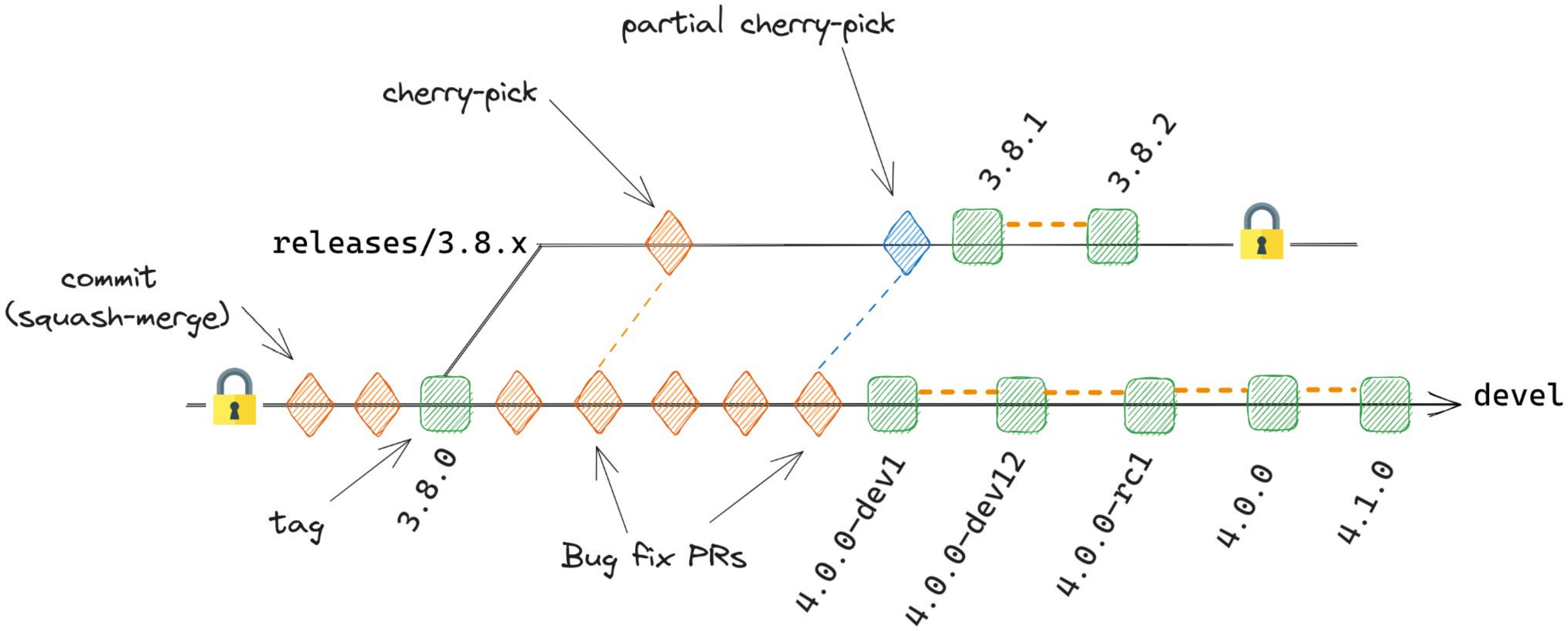
- Documentation builds automatically:
  - Merge to devel branch: www.avd.sh/en**/devel/**
  - Production release tag: www.avd.sh/en**/stable/**
  - Specific release: www.avd.sh/en**/vX.X.X/**

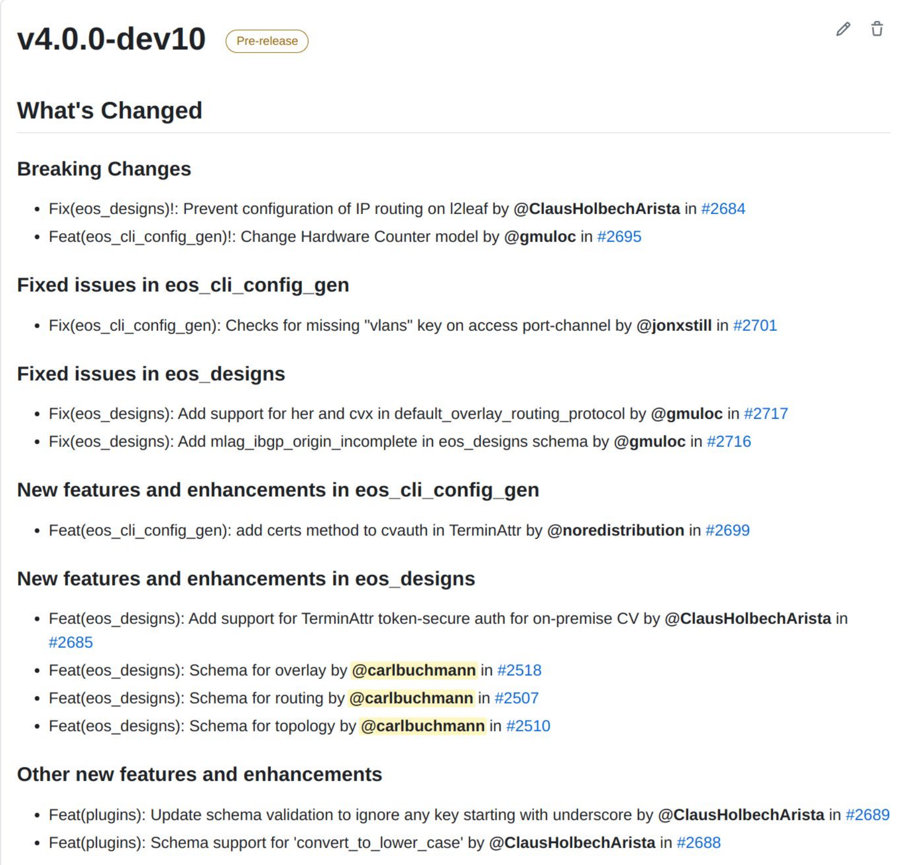
GitHub ↔ Read *the* Docs

AVD AutoTest



Release Strategy





v4.0.0-dev10

Pre-release

What's Changed

Breaking Changes

- Fix(eos_designs)!: Prevent configuration of IP routing on l2leaf by **@ClausHolbechArista** in #2684
- Feat(eos_cli_config_gen)!: Change Hardware Counter model by **@gmuloc** in #2695

Fixed issues in eos_cli_config_gen

- Fix(eos_cli_config_gen): Checks for missing "vlans" key on access port-channel by **@jonxstill** in #2701

Fixed issues in eos_designs

- Fix(eos_designs): Add support for her and cvx in default_overlay_routing_protocol by **@gmuloc** in #2717
- Fix(eos_designs): Add mlag_ibgp_origin_incomplete in eos_designs schema by **@gmuloc** in #2716

New features and enhancements in eos_cli_config_gen

- Feat(eos_cli_config_gen): add certs method to cvauth in TerminAttr by **@noredistribution** in #2699

New features and enhancements in eos_designs

- Feat(eos_designs): Add support for TerminAttr token-secure auth for on-premise CV by **@ClausHolbechArista** in #2685
- Feat(eos_designs): Schema for overlay by **@carlbuchmann** in #2518
- Feat(eos_designs): Schema for routing by **@carlbuchmann** in #2507
- Feat(eos_designs): Schema for topology by **@carlbuchmann** in #2510

Other new features and enhancements

- Feat(plugins): Update schema validation to ignore any key starting with underscore by **@ClausHolbechArista** in #2689
- Feat(plugins): Schema support for 'convert_to_lower_case' by **@ClausHolbechArista** in #2688

Release to RedHat Automation Hub

CERTIFIED

Red Hat Ansible Certified Content
available through console.redhat.com

Establishes integration with partner platforms, typically in the form of modules. It provides the integration for what is being automated.

✓ Subscription required

✓ Signed content for support of a secure software supply chain

VALIDATED

Ansible validated content
pre-loaded into private automation hub

An expert-led path for performing operations/tasks in the partner's platform and sometimes referencing the integration. It provides guidance for how to optimize the solution based on subject-matter expertise.

Typically in the form of roles and playbooks.

✓ Subscription required

✓ Signed content for support of a secure software supply chain

Thank you

- Documentation
- GitHub repository
- Arista community examples
- Contact

Red Hat

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

- facebook.com/ansibleautomation
- twitter.com/ansible
- linkedin.com/company/ansible/
- youtube.com/user/RedHatVideos