

# Ansible in a Devcontainer

Entire arista.avd ecosystem in a sealed bottle

Petr Ankudinov  
Patrick Mathy

2023



## \$ whoami

- Petr Ankudinov [github.com/ankudinov](https://github.com/ankudinov)
  - Advanced Services Engineer at Arista Networks
  - Over 20 years of experience in IT with a bit of everything
  - ACE: L5, CCIE 37521
  - Passionate DC and network automation engineer
  - Daily (and nightly) user of Ansible, VSCode and more
- Patrick Mathy
  - Arista Systems Engineering at Arista Networks
  - Networking around since 2016
  - ACE: L5, CCIE 57751
  - R&S, DC, Python, Ansible, Terraform, DevNet



# Agenda

- Ansible AVD collection overview
- Common challenges when building Ansible environment for network automation
- Why devcontainers?
- Pre-building a devcontainer with [arista.avd](#), docker-in-docker and Containerlab using Github [devcontainers/ci@v0.3](#) action.
- How to run the container on any machine (with docker run or as devcontainer) or Github Codespaces



# Credits and References

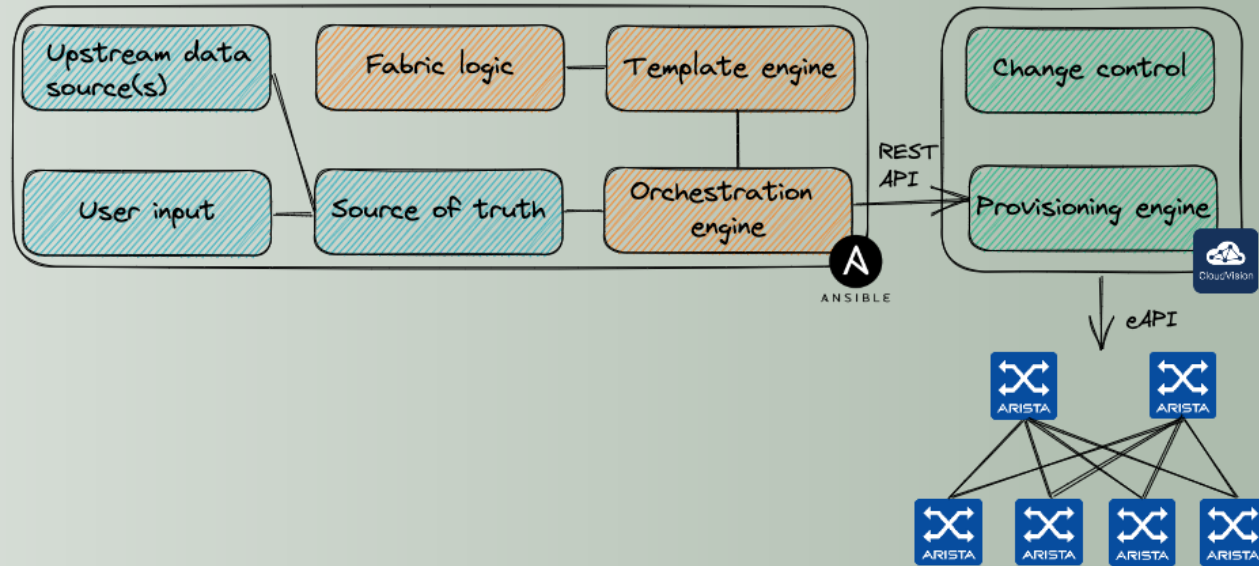
This repository is based on many awesome open source repositories and some free/commercial Github features:

- [VS Code](#)
- [DevContainers](#)
- [Marp](#)
- [Excalidraw VS Code Plugin](#)
- [Github Actions](#)
- [Github Pages](#)
- [Github Codespaces](#)
- [Carbon](#)
- And many more...

All photos are taken from [Pexels](#) and [Unsplash](#). Excellent free stock photos resources. It's not possible to reference every author individually, but their work is highly appreciated.

# What is Ansible AVD?

- **AVD** stands for Arista Validated Design as it was based on the [EVPN Deployment Guide](#)
- A very successful community project used to deploy EVPN based Data Center fabrics
  - Over [200 stars on Github](#) and 79 contributors as of Sep 2023
  - The most active Arista collection on [Ansible Galaxy](#)
- High level workflow:
  - Define abstracted group/host vars using AVD data model
  - Generate low level device specific variables (aka structured configs)
  - Parse templates, build plain text configs
  - Deliver configs to network devices using Ansible `arista.eos.eos_config`



# Running Ansible in a Container

- The old story of "it works on my machine":
  - Different versions of Python and Ansible
  - Dependencies
  - Interpreter path issues
  - The famous very-very-very-**VERY** verbose only to find out that:

The error appears to be, but may be elsewhere (c) Ansible 😊

The error handling and input validation is a very significant part of the `ansible.avd` collection.

- Containers help to solve the issues above. But bring new challenges and not always easy to build and use.



# What It Takes to Build a Good Container?

- Craft a Dockerfile with some essentials.
- Add a non-root user, as root breaks permissions, breaks Ansible and ruins your work-life balance 😎.
- Match user ID inside and outside of the container.  
Some operating systems like RHEL and the family are very strict about it. This is not a trivial task.
- Create an entrypoint.
- Take care of transferring Git credentials, keys, etc. into the container (if it's interactive).
- Think about security and maintaining the container repository.
- ... and it has to be multi-platform: amd64 and arm64 as a minimum.

And now convince someone to run it. 🧑 ➡️

```
docker run --rm -it \  
    --network host \  
    --pid="host" \  
    -w $(CURRENT_DIR) \  
    -v $(CURRENT_DIR):$(CURRENT_DIR) \  
    -e AVD_GIT_USER="$(shell git config --get user.name)" \  
    -e AVD_GIT_EMAIL="$(shell git config --get user.email)" \  
    $(AVD_CONTAINER_IMAGE) || true
```