Ansible in a Devcontainer

```
Entire arista.avd ecosystem in a sealed bottle

Petr Ankudinov
Patrick Mathy
                                          2023
```

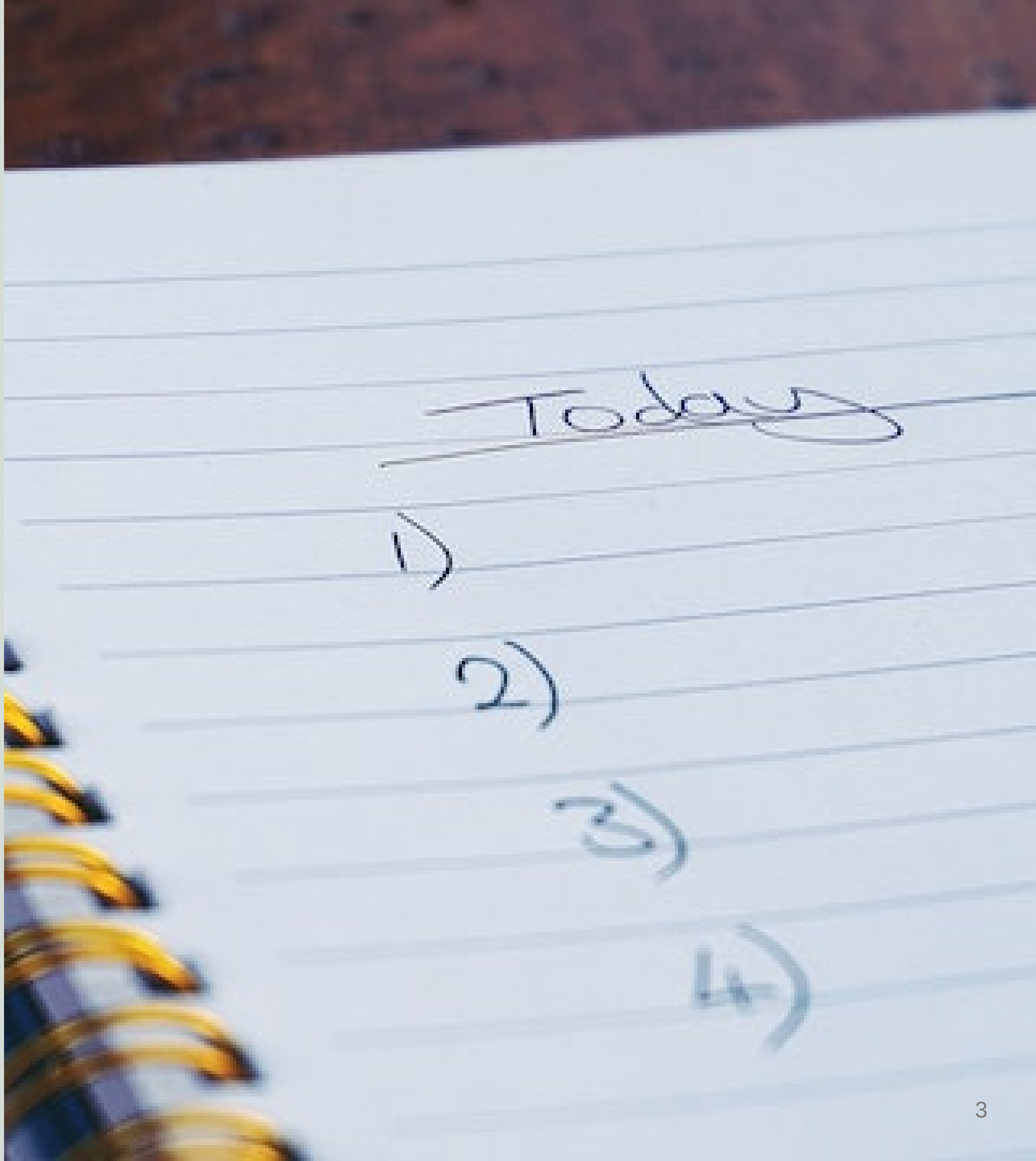
$ whoami

- Petr Ankudinov 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

```
Tech level: intermediate
  know some Ansible, VSCode,
  containers, Github, etc.
```



Credits and References

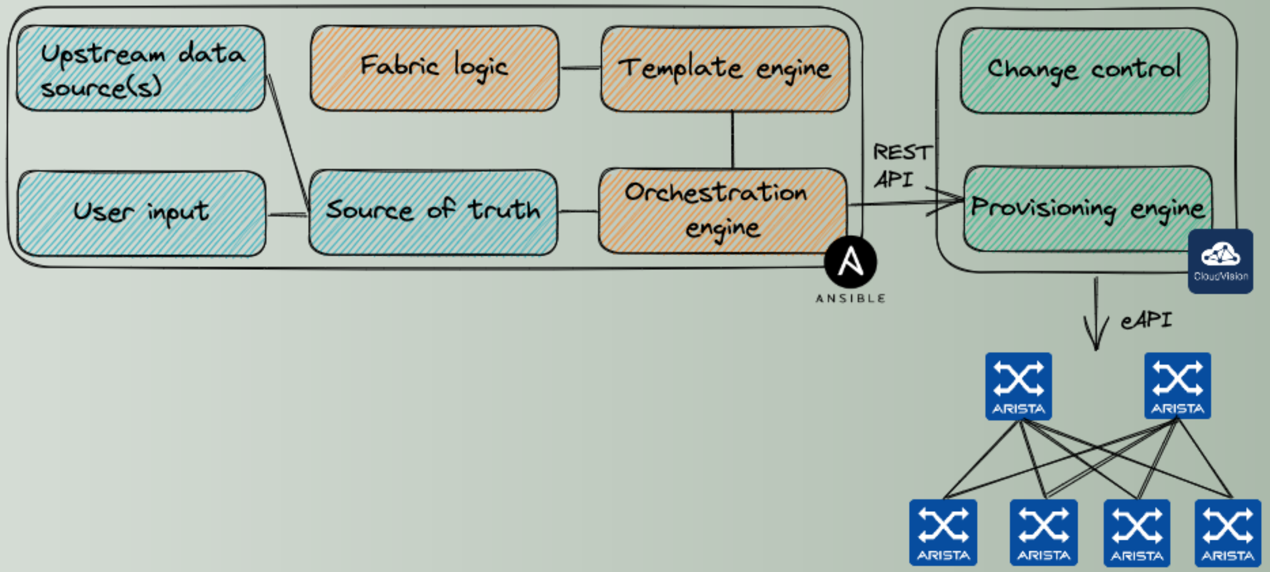
This repository is based on many awesome open source repositories and some free/commercial Github features:

- 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_config`



Running Ansible in a Container

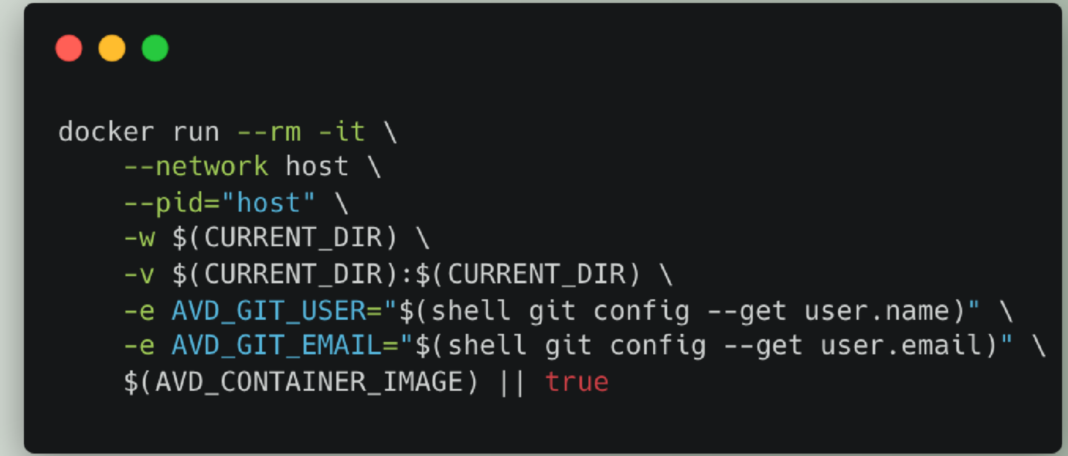
- The old story of "it works on my machine":
  - Different versions of Python and Ansible
  - Dependencies
  - Interpreter path issues
  - The famous very-very-very-**VERY** verbose only to find out that: `The error appears to be, but may be elsewhere` (c) Ansible 😅

    > 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. 🥷 ➡️

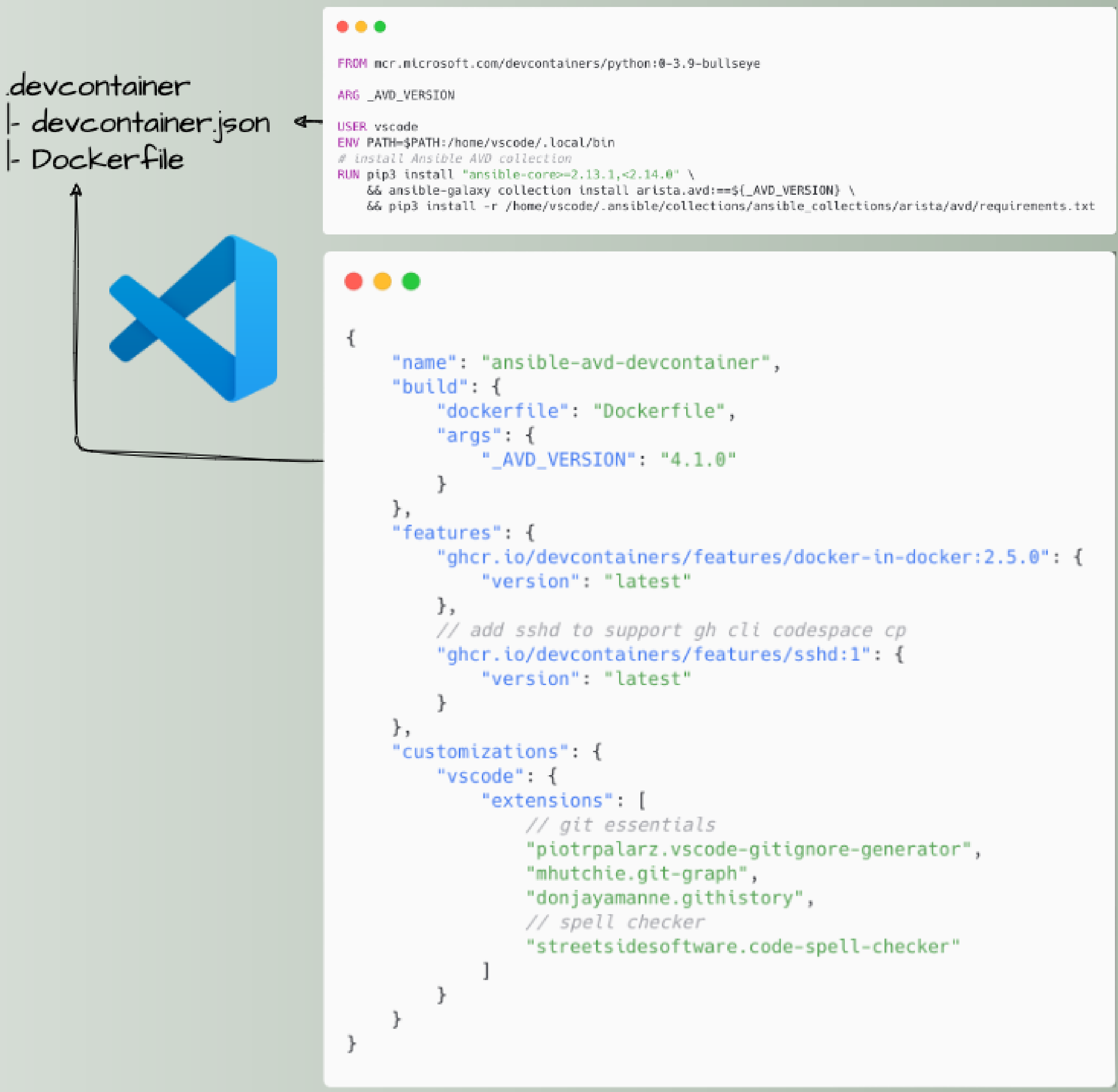


```
docker run --rm -it \
    --network host \
    --pid="host" \
    -w $(CURRENT_DIR) \
    -v $(CURRENT_DIR):$(CURRENT_DIR) \
    -e AVD_GIT_USER="$(shell git config --get user.name)" \
    -e AVD_GIT_EMAIL="$(shell git config --get user.email)" \
    $(AVD_CONTAINER_IMAGE) || true
```

Dev Container - A Better Container

- A Dev Container is a container used as a fully featured development environment. Dev containers can be run locally or remotely, in a private or public cloud, in a variety of supporting tools and editors.
- Dev Container Specification was started by Microsoft and has strong community support.
- Dev Containers are powered by:
  - Prebuilt images
  - Features

> **Dev Container features** enable complex functionality at the cost a few lines added to `devcontainer.json`

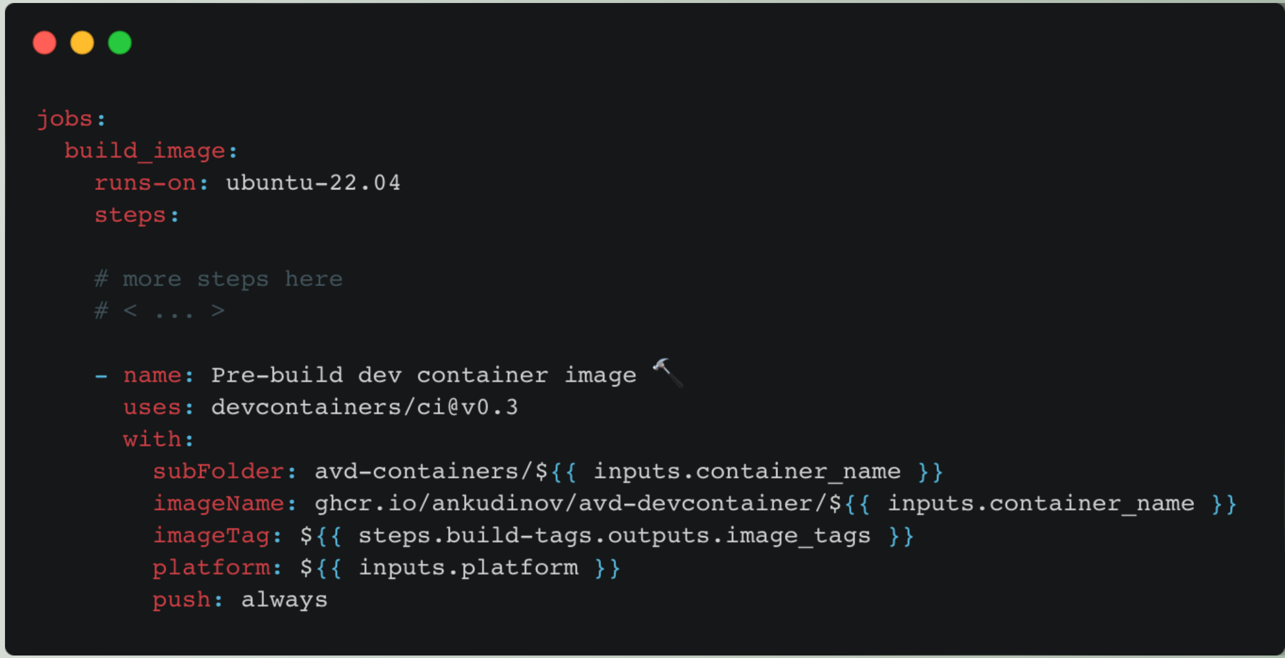


```
FROM mcr.microsoft.com/devcontainers/python:0-3.9-bullseye

ARG _AVD_VERSION

USER vscode
ENV PATH=$PATH:/home/vscode/.local/bin
# install Ansible AVD collection
RUN pip3 install "ansible-core>=2.13.1,<2.14.0" \
    && ansible-galaxy collection install arista.avd:==${_AVD_VERSION} \
    && pip3 install -r /home/vscode/.ansible/collections/ansible_collections/arista/avd/requirements.txt
```

```
{
    "name": "ansible-avd-devcontainer",
    "build": {
        "dockerfile": "Dockerfile",
        "args": {
            "_AVD_VERSION": "4.1.0"
        }
    },
    "features": {
        "ghcr.io/devcontainers/features/docker-in-docker:2.5.0": {
            "version": "latest"
        },
        // add sshd to support gh cli codespace cp
        "ghcr.io/devcontainers/features/sshd:1": {
            "version": "latest"
        }
    },
    "customizations": {
        "vscode": {
            "extensions": [
                // git essentials
                "piotrpalarz.vscode-gitignore-generator",
                "mhutchie.git-graph",
                "donjayamanne.githistory",
                // spell checker
                "streetsidesoftware.code-spell-checker"
            ]
        }
    }
}
```

Prebuilt Dev Containers

- Building a dev container locally may not be optimal and increases the risk of changing dependencies.
- You can pre-built your own dev container and upload to any container registry.
- One of the best combos:
  - Github Container Registry
  - devcontainers/ci@v0.3 action

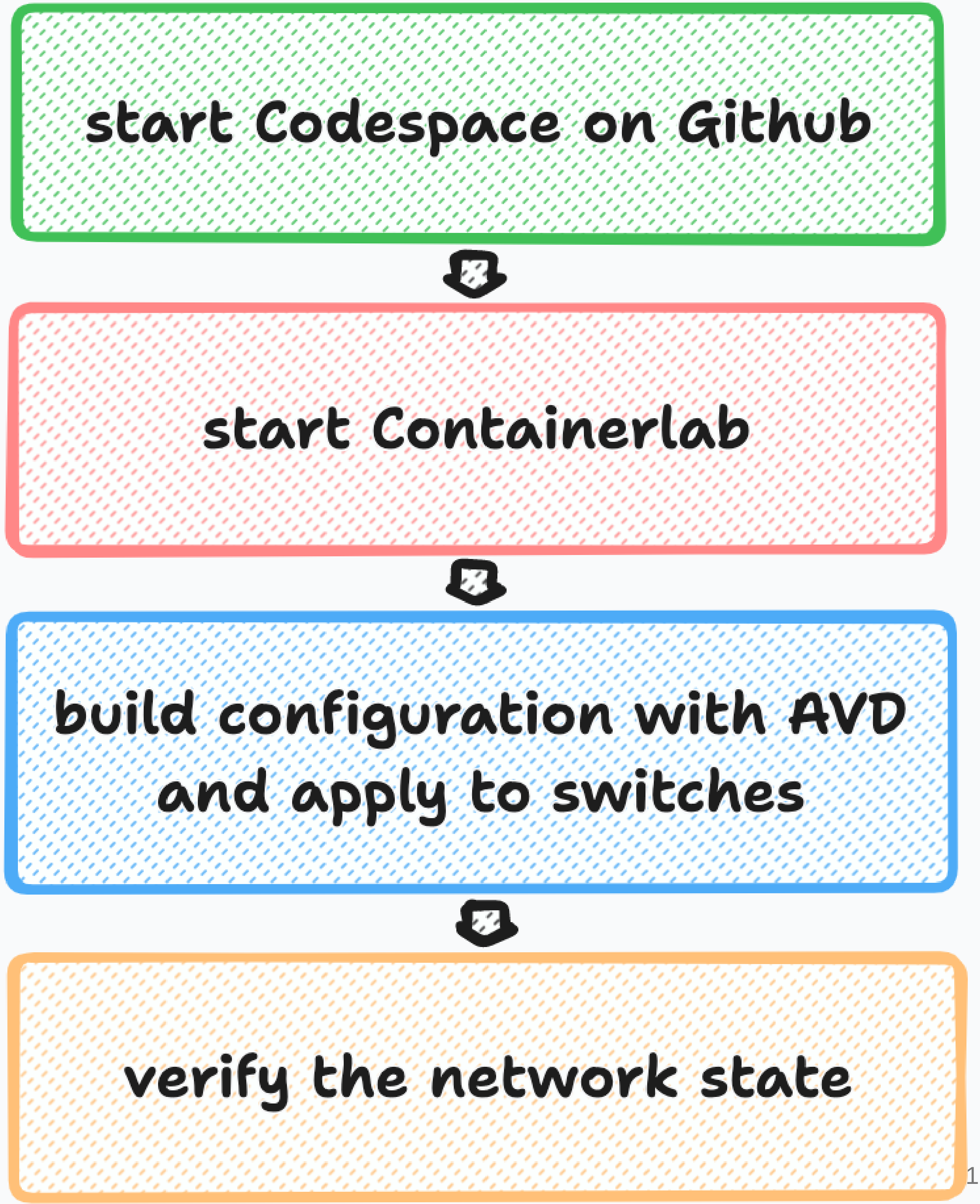


```yaml
jobs:
  build_image:
    runs-on: ubuntu-22.04
    steps:

    # more steps here
    # < ... >

    - name: Pre-build dev container image 🔨
      uses: devcontainers/ci@v0.3
      with:
        subFolder: avd-containers/${{ inputs.container_name }}
        imageName: ghcr.io/ankudinov/avd-devcontainer/${{ inputs.container_name }}
        imageTag: ${{ steps.build-tags.outputs.image_tags }}
        platform: ${{ inputs.platform }}
        push: always
```

ARISTA

The Demo

- The demo is showing a single use case of building a functional EVPN lab in a dev container with Ansible AVD
- The use cases are endless
- Dev containers are not intended to be used in prod theoretically, but in certain cases this can be very acceptable

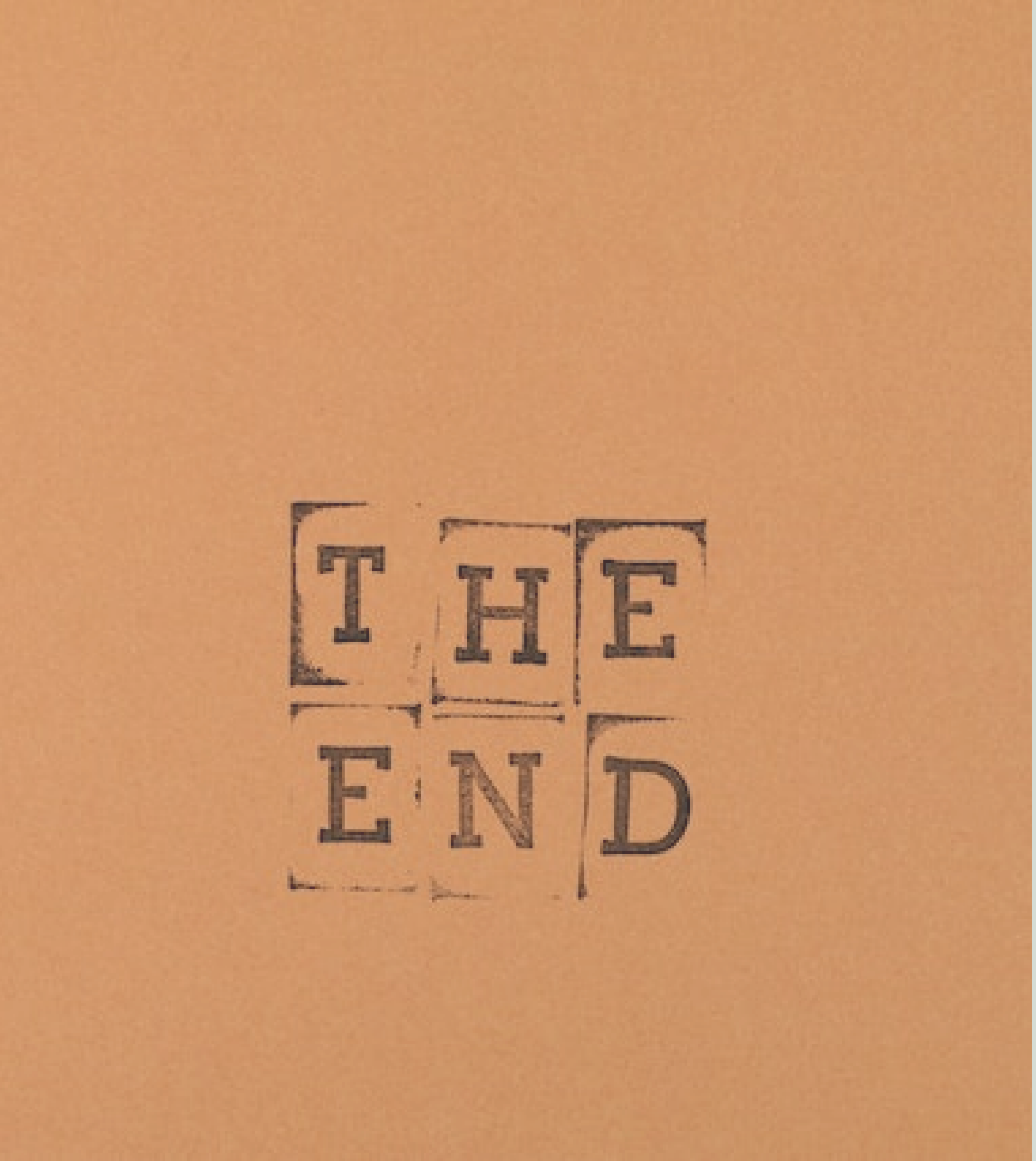


start Codespace on Github

start Containerlab

build configuration with AVD and apply to switches

verify the network state



Q&A

ARISTA