LF NETWORKING
Developer & Testing Forum

Open discussion on Test-Infra

Victor Morales

Samsung

https://lfnetworking.org

Agenda

- Project overview
  - Miscellaneous folders
  - e2e folder
    - Pre-bake VM image
    - Cloud CI management
    - Sandbox provision
- Areas for improvement
  - Extract Ansible Roles
  - Implement KUTTL Framework
  - Use Doc Detective tool
  - Replace MetalLB for KinD cloud provider
  - Use Ephemeral Docker image registry

Project overview

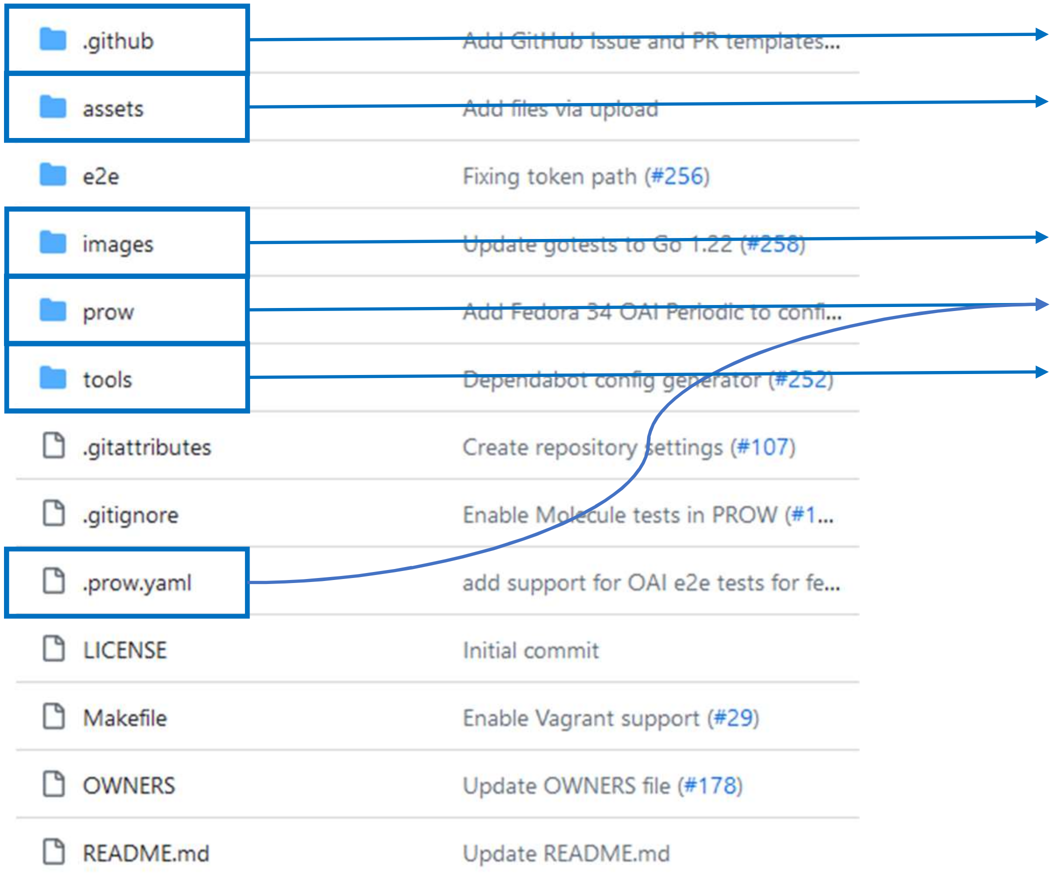
Overview

Test Infrastructure provide tools and scripts to guarantee the proper functionality of Nephio Components

- Provides scripts to install requirements used by the Test Cases (ansible bootstrap role).
- Installs Nephio components (ansible install role).
- Contains a kpt Ansible module.
- Provisions a local/remote Nephio Sandbox through a Vagrantfile.
- Provides Terraform scripts utilized by CI.
- Stores Dockerfiles for CI artifacts.
- Contains bash scripts used during the execution of the End-To-End testing.

https://github.com/nephio-project/test-infra/

Miscellaneous folders



.github	Add GitHub Issue and PR templates...
assets	Add files via upload
e2e	Fixing token path (#256)
images	Update gotests to Go 1.22 (#258)
prow	Add Fedora 34 OAI Periodic to confi...
tools	Dependabot config generator (#252)
.gitattributes	Create repository settings (#107)
.gitignore	Enable Molecule tests in PROW (#1...
.prow.yaml	add support for OAI e2e tests for fe...
LICENSE	Initial commit
Makefile	Enable Vagrant support (#29)
OWNERS	Update OWNERS file (#178)
README.md	Update README.md

GitHub PR templates (eventually GitHub actions)

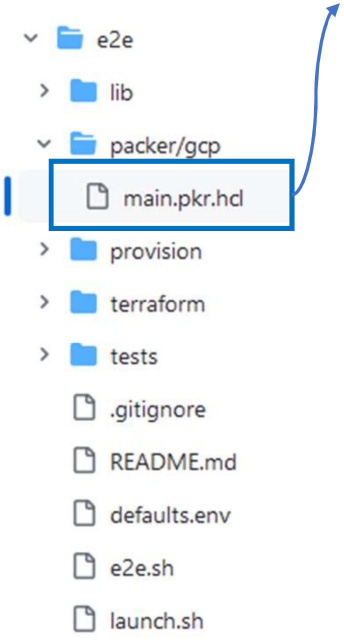
Images used by the Wiki

Docker image definitions for CI artifacts

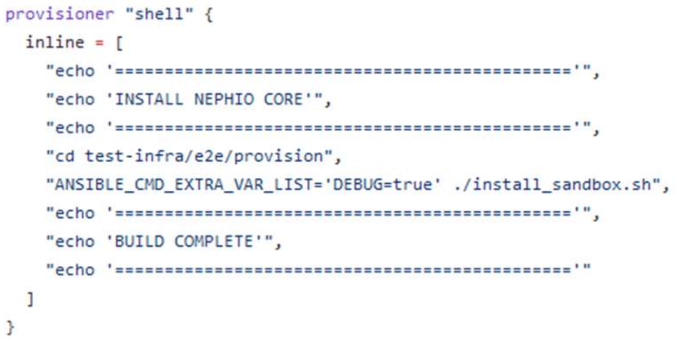
Prow configuration

Python scripts for dependabot and release GitHub actions

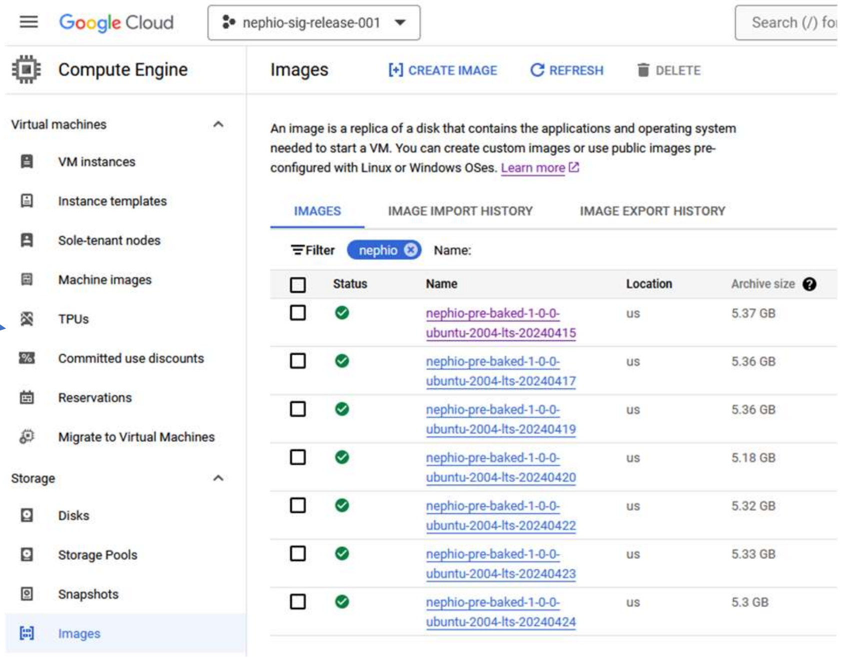
e2e folder – Pre-bake VM image



```
⌄ e2e
  › lib
  ⌄ packer/gcp
      main.pkr.hcl
  › provision
  › terraform
  › tests
    .gitignore
    README.md
    defaults.env
    e2e.sh
    launch.sh
```



```
provisioner "shell" {
  inline = [
    "echo '============================================='",
    "echo 'INSTALL NEPHIO CORE'",
    "echo '============================================='",
    "cd test-infra/e2e/provision",
    "ANSIBLE_CMD_EXTRA_VAR_LIST='DEBUG=true' ./install_sandbox.sh",
    "echo '============================================='",
    "echo 'BUILD COMPLETE'",
    "echo '============================================='"
  ]
}
```



Images

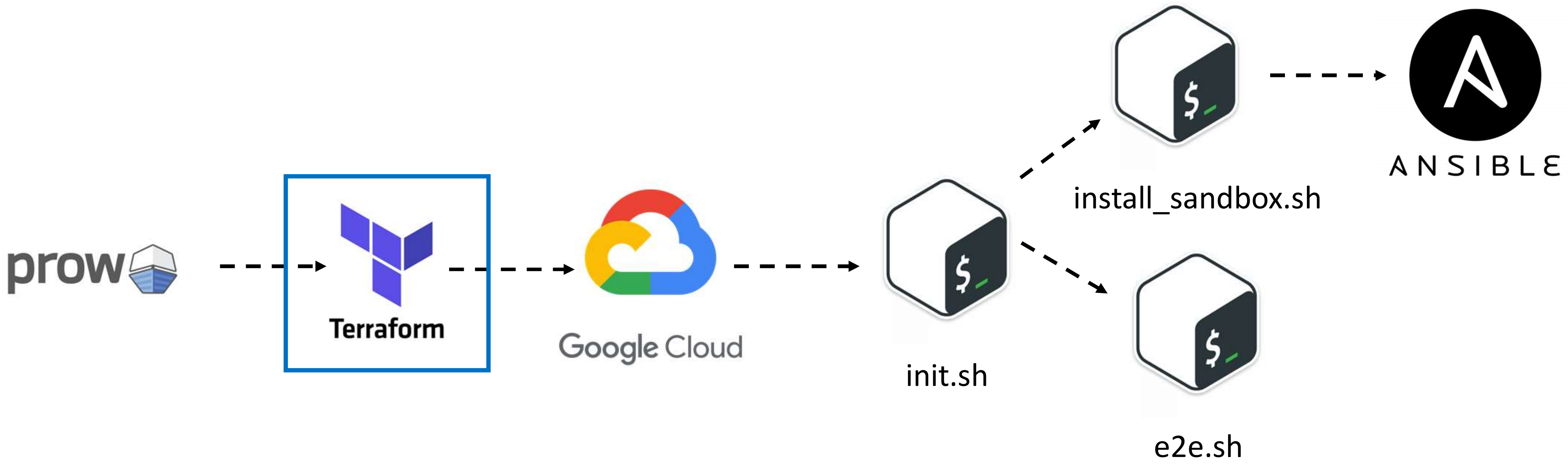
An image is a replica of a disk that contains the applications and operating system needed to start a VM. You can create custom images or use public images pre-configured with Linux or Windows OSes. Learn more

□	Status	Name	Location	Archive size
□	✓	nephio-pre-baked-1-0-0-ubuntu-2004-lts-20240415	us	5.37 GB
□	✓	nephio-pre-baked-1-0-0-ubuntu-2004-lts-20240417	us	5.36 GB
□	✓	nephio-pre-baked-1-0-0-ubuntu-2004-lts-20240419	us	5.36 GB
□	✓	nephio-pre-baked-1-0-0-ubuntu-2004-lts-20240420	us	5.18 GB
□	✓	nephio-pre-baked-1-0-0-ubuntu-2004-lts-20240422	us	5.32 GB
□	✓	nephio-pre-baked-1-0-0-ubuntu-2004-lts-20240423	us	5.33 GB
□	✓	nephio-pre-baked-1-0-0-ubuntu-2004-lts-20240424	us	5.3 GB

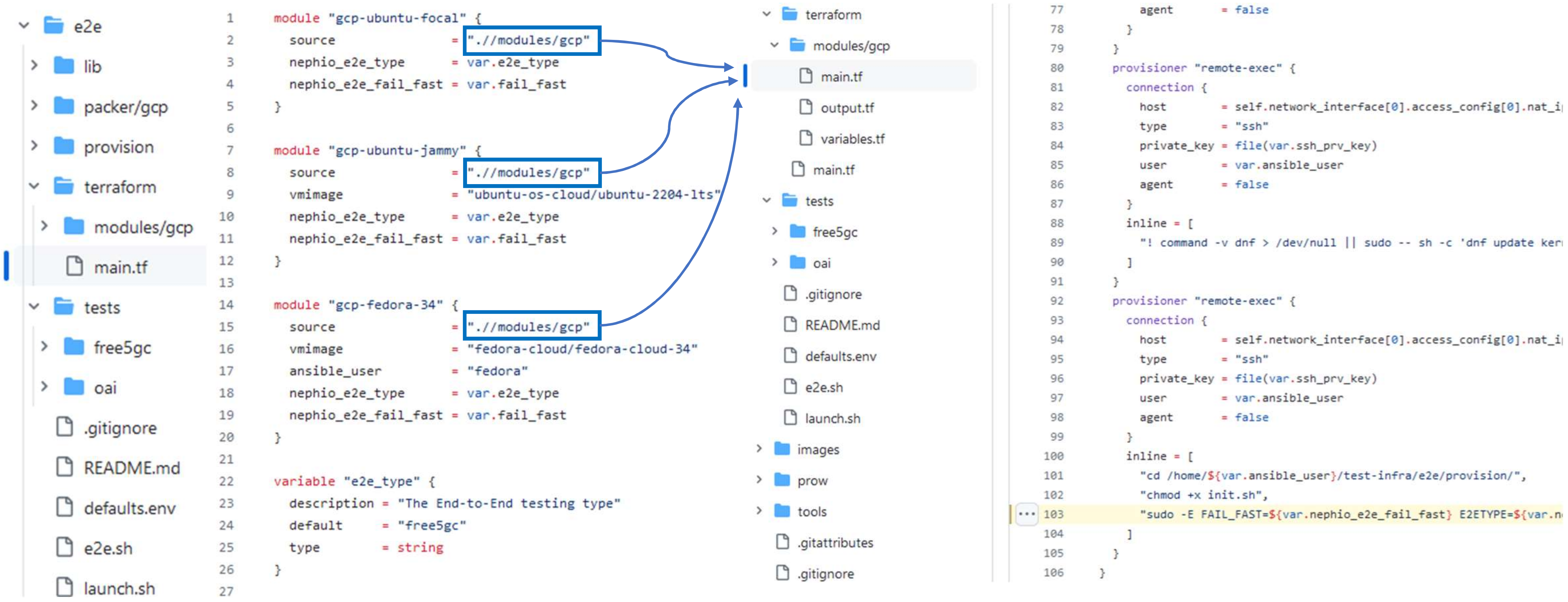
Call for action:

- Expose “nephio-pre-baked-*” images for consumption
- Implement cleanup procedure to avoid cloud costs

CI Workflow overview



e2e folder – Cloud CI mgmt.

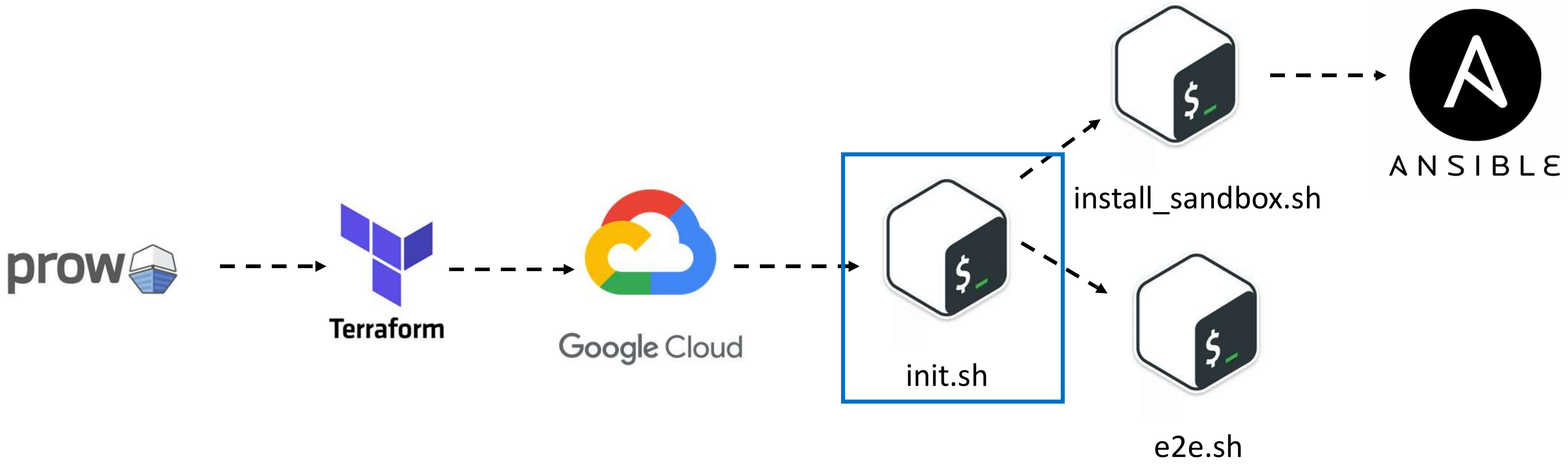


```
module "gcp-ubuntu-focal" {
  source             = "./modules/gcp"
  nephio_e2e_type    = var.e2e_type
  nephio_e2e_fail_fast = var.fail_fast
}

module "gcp-ubuntu-jammy" {
  source             = "./modules/gcp"
  vmimage            = "ubuntu-os-cloud/ubuntu-2204-lts"
  nephio_e2e_type    = var.e2e_type
  nephio_e2e_fail_fast = var.fail_fast
}

module "gcp-fedora-34" {
  source             = "./modules/gcp"
  vmimage            = "fedora-cloud/fedora-cloud-34"
  ansible_user       = "fedora"
  nephio_e2e_type    = var.e2e_type
  nephio_e2e_fail_fast = var.fail_fast
}

variable "e2e_type" {
  description = "The End-to-End testing type"
  default     = "free5gc"
  type        = string
}
```

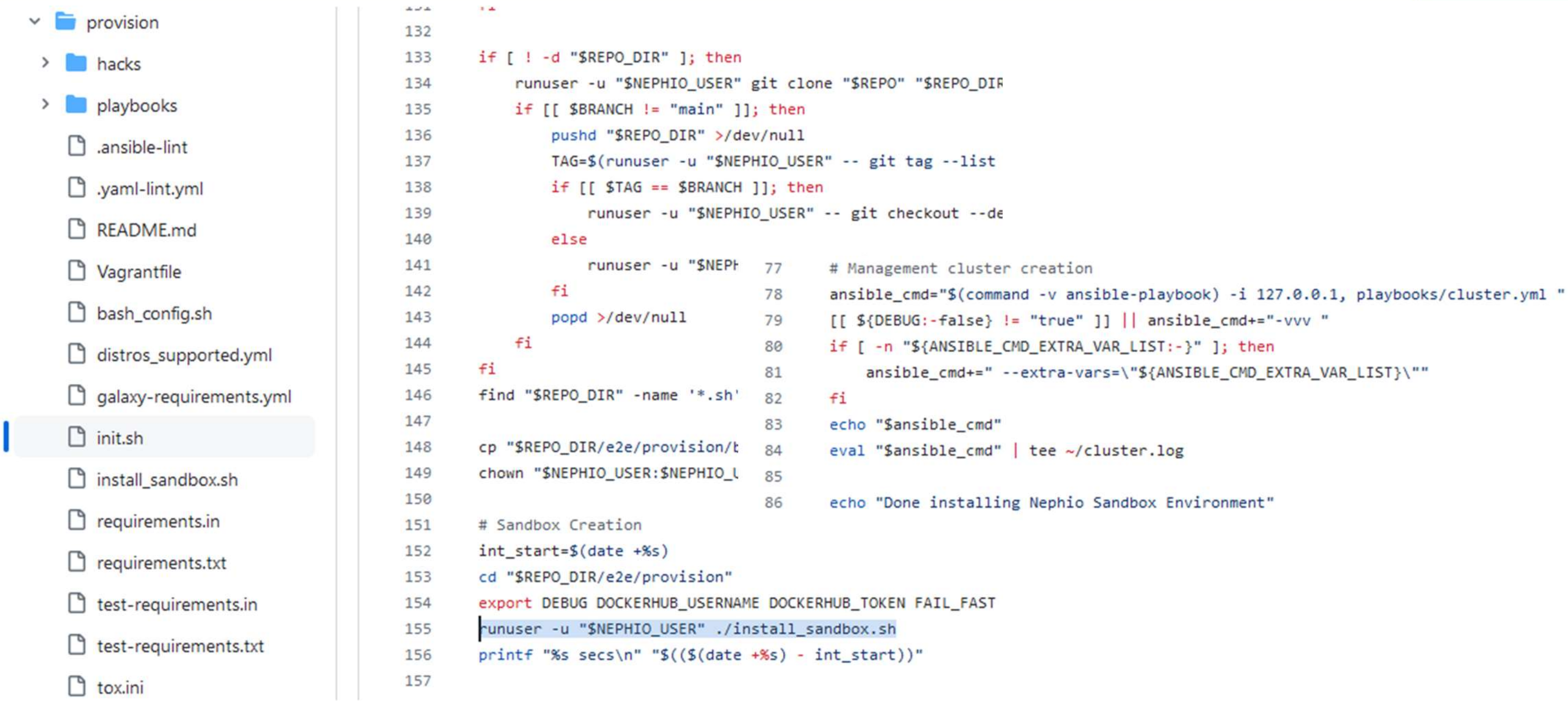
```
        agent       = false
      }
    }
    provisioner "remote-exec" {
      connection {
        host        = self.network_interface[0].access_config[0].nat_i
        type        = "ssh"
        private_key = file(var.ssh_prv_key)
        user        = var.ansible_user
        agent       = false
      }
      inline = [
        "! command -v dnf > /dev/null || sudo -- sh -c 'dnf update ker
      ]
    }
    provisioner "remote-exec" {
      connection {
        host        = self.network_interface[0].access_config[0].nat_i
        type        = "ssh"
        private_key = file(var.ssh_prv_key)
        user        = var.ansible_user
        agent       = false
      }
      inline = [
        "cd /home/${var.ansible_user}/test-infra/e2e/provision/",
        "chmod +x init.sh",
        "sudo -E FAIL_FAST=${var.nephio_e2e_fail_fast} E2ETYPE=${var.n
      ]
    }
  }
```

https://github.com/nephio-project/test-infra/blob/main/e2e/terraform/modules/gcp/main.tf#L103

CI Workflow overview



e2e folder – Sandbox provision (1/3)



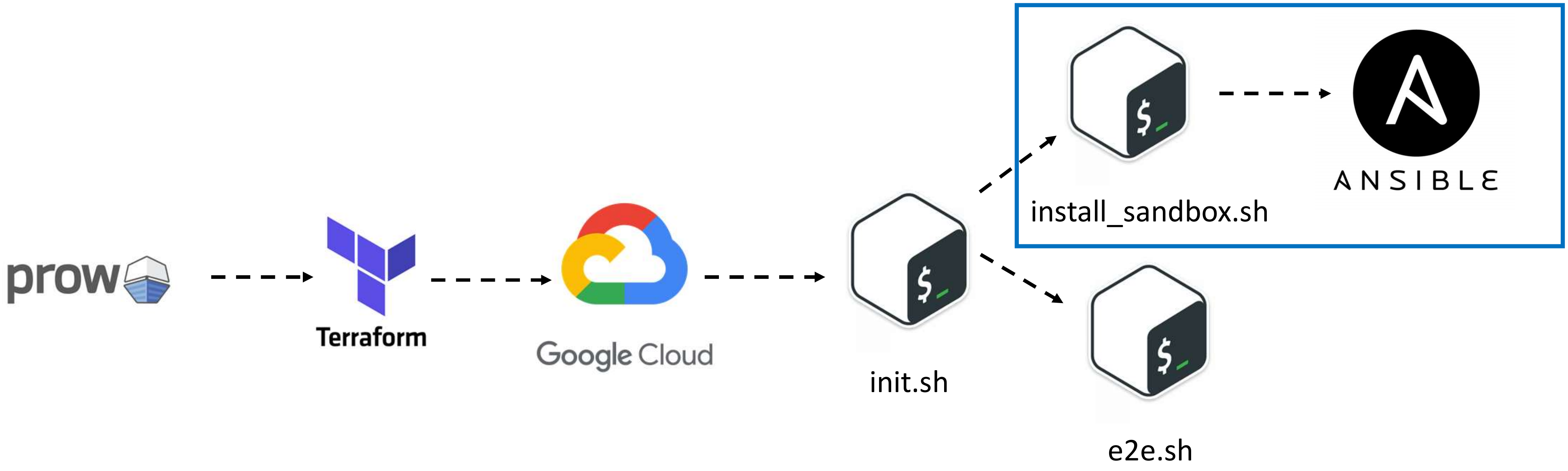
- provision
  - hacks
  - playbooks
  - .ansible-lint
  - .yaml-lint.yml
  - README.md
  - Vagrantfile
  - bash_config.sh
  - distros_supported.yml
  - galaxy-requirements.yml
  - init.sh
  - install_sandbox.sh
  - requirements.in
  - requirements.txt
  - test-requirements.in
  - test-requirements.txt
  - tox.ini

```
if [ ! -d "$REPO_DIR" ]; then
    runuser -u "$NEPHIO_USER" git clone "$REPO" "$REPO_DIR
    if [[ $BRANCH != "main" ]]; then
        pushd "$REPO_DIR" >/dev/null
        TAG=$(runuser -u "$NEPHIO_USER" -- git tag --list
        if [[ $TAG == $BRANCH ]]; then
            runuser -u "$NEPHIO_USER" -- git checkout --de
        else
            runuser -u "$NEPH
        fi
        popd >/dev/null
    fi
fi
find "$REPO_DIR" -name '*.sh'

cp "$REPO_DIR/e2e/provision/t
chown "$NEPHIO_USER:$NEPHIO_U

# Sandbox Creation
int_start=$(date +%s)
cd "$REPO_DIR/e2e/provision"
export DEBUG DOCKERHUB_USERNAME DOCKERHUB_TOKEN FAIL_FAST
runuser -u "$NEPHIO_USER" ./install_sandbox.sh
printf "%s secs\n" "$(($(date +%s) - int_start))"
```

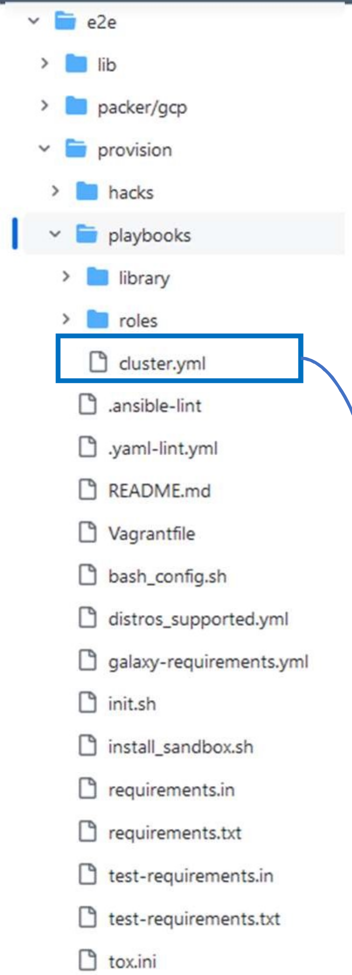
```
# Management cluster creation
ansible_cmd="$(command -v ansible-playbook) -i 127.0.0.1, playbooks/cluster.yml "
[[ ${DEBUG:-false} != "true" ]] || ansible_cmd+="-vvv "
if [ -n "${ANSIBLE_CMD_EXTRA_VAR_LIST:-}" ]; then
    ansible_cmd+=" --extra-vars=\"${ANSIBLE_CMD_EXTRA_VAR_LIST}\""
fi
echo "$ansible_cmd"
eval "$ansible_cmd" | tee ~/cluster.log

echo "Done installing Nephio Sandbox Environment"
```

https://github.com/nephio-project/test-infra/blob/main/e2e/provision/install_sandbox.sh

CI Workflow overview

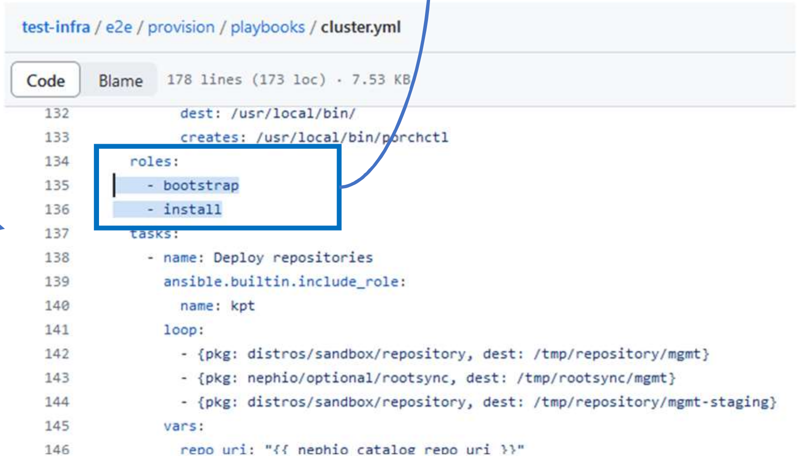


e2e folder – Sandbox provision (2/3)

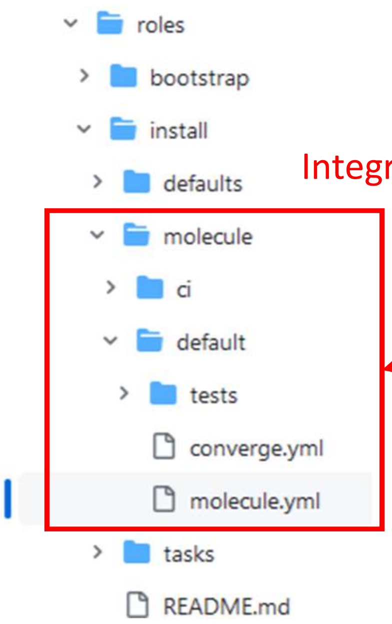


Provides two main Ansible roles:

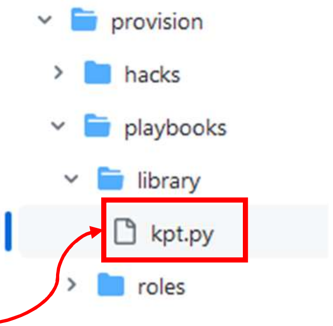
1. bootstrap – Validate host requirements, loads kernel modules, creates a KinD management cluster, installs gitea, resource-backend, cluster-api, cert-manager and metalLB packages in parallel.
2. install – Provides a local or **remote** installation of the Nephio components in **parallel**.


test-infra / e2e / provision / playbooks / cluster.yml
Code
Blame
178 lines (173 loc) · 7.53 KB

```yaml
          dest: /usr/local/bin/
          creates: /usr/local/bin/porchctl
    roles:
      - bootstrap
      - install
    tasks:
      - name: Deploy repositories
        ansible.builtin.include_role:
          name: kpt
        loop:
          - {pkg: distros/sandbox/repository, dest: /tmp/repository/mgmt}
          - {pkg: nephio/optional/rootsync, dest: /tmp/rootsync/mgmt}
          - {pkg: distros/sandbox/repository, dest: /tmp/repository/mgmt-staging}
        vars:
          repo_uri: "{{ nephio_catalog_repo_uri }}"
```

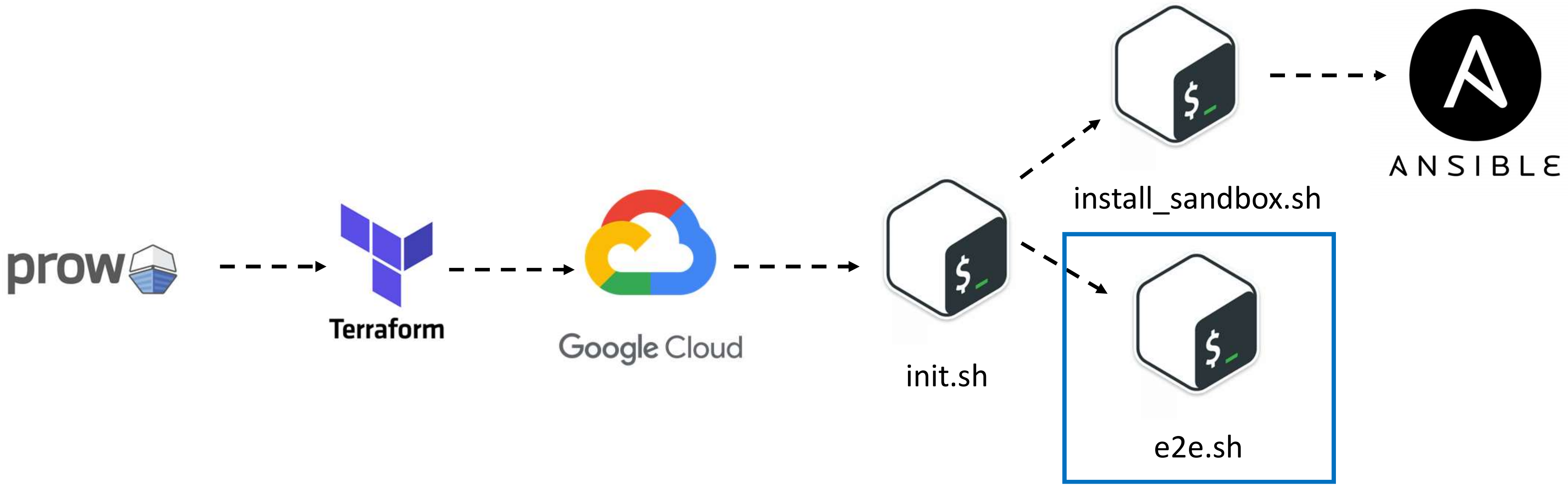


Integration testing

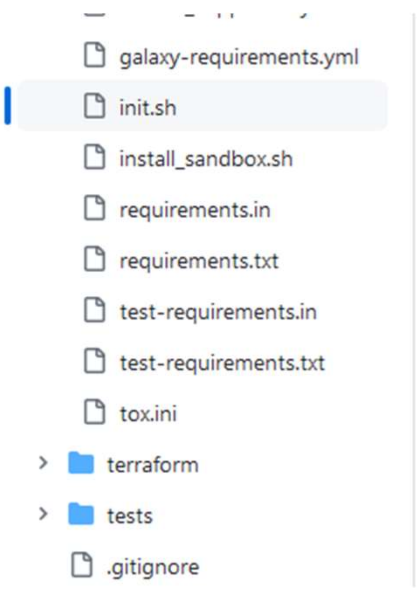


kpt ansible module

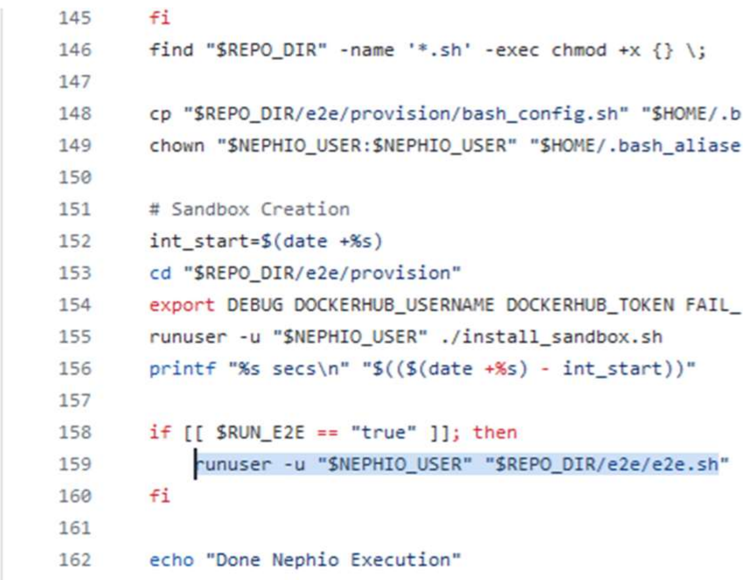
CI Workflow overview



e2e folder – Sandbox provision (3/3)

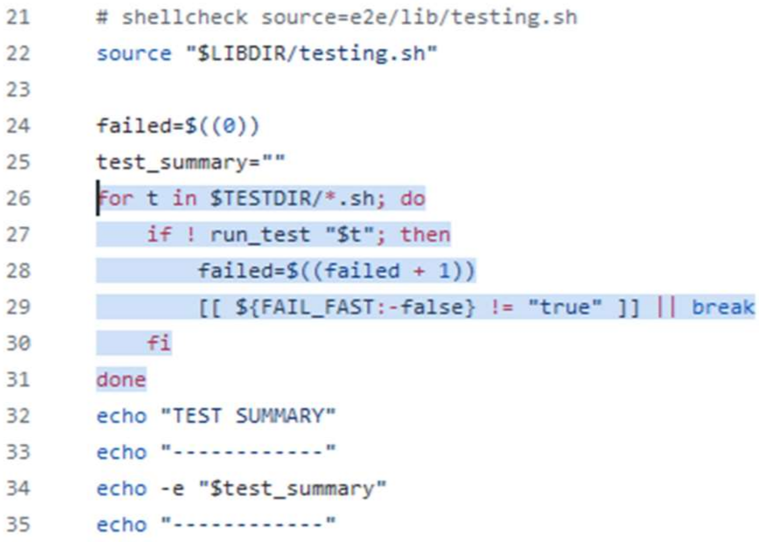


- galaxy-requirements.yml
- init.sh
- install_sandbox.sh
- requirements.in
- requirements.txt
- test-requirements.in
- test-requirements.txt
- tox.ini
- terraform
- tests
- .gitignore

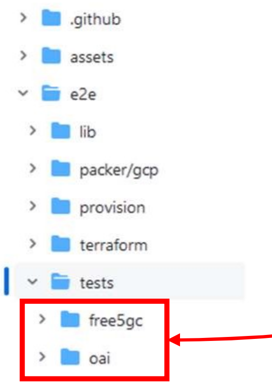


```
fi
find "$REPO_DIR" -name '*.sh' -exec chmod +x {} \;

cp "$REPO_DIR/e2e/provision/bash_config.sh" "$HOME/.b
chown "$NEPHIO_USER:$NEPHIO_USER" "$HOME/.bash_aliase

# Sandbox Creation
int_start=$(date +%s)
cd "$REPO_DIR/e2e/provision"
export DEBUG DOCKERHUB_USERNAME DOCKERHUB_TOKEN FAIL_
runuser -u "$NEPHIO_USER" ./install_sandbox.sh
printf "%s secs\n" "$(($(date +%s) - int_start))"

if [[ $RUN_E2E == "true" ]]; then
    runuser -u "$NEPHIO_USER" "$REPO_DIR/e2e/e2e.sh"
fi

echo "Done Nephio Execution"
```

https://github.com/nephio-project/test-infra/blob/main/e2e/e2e.sh



```
# shellcheck source=e2e/lib/testing.sh
source "$LIBDIR/testing.sh"

failed=$((0))
test_summary=""
for t in $TESTDIR/*.sh; do
    if ! run_test "$t"; then
        failed=$((failed + 1))
        [[ ${FAIL_FAST:-false} != "true" ]] || break
    fi
done
echo "TEST SUMMARY"
echo "------------"
echo -e "$test_summary"
echo "------------"
```



- .github
- assets
- e2e
  - lib
  - packer/gcp
  - provision
  - terraform
  - tests
    - free5gc
    - oai

Test cases resources

LF NETWORKING
LFN Developer & Testing Forum

Areas for improvement

Checkpoint/Restore KinD containers

Provide Nephio management checkpoints for fasters startups

https://criu.org

```
$ docker checkpoint create kind-nephio 20240105
```

https://packages.ubuntu.com/jammy/criu
https://packages.fedoraproject.org/pkgs/criu/criu/

Pros	Cons
Avoid cold starts	Experimental feature
Ubuntu 22.04+ and Fedora38+ packages	
Reutilization of code	

Extract Ansible Roles

The following Ansible role needs their own repository:

- https://github.com/nephio-project/test-infra/tree/main/e2e/provision/playbooks/roles/bootstrap
- https://github.com/nephio-project/test-infra/tree/main/e2e/provision/playbooks/roles/install
- https://github.com/nephio-project/test-infra/tree/main/e2e/provision/playbooks/roles/kpt
- https://github.com/nephio-project/test-infra/tree/main/e2e/provision/playbooks/roles/upgrade

Pros	Cons
Major visibility	kpt Ansible module dependency
Faster development cycles	Complex troubleshooting
Reutilization of code	

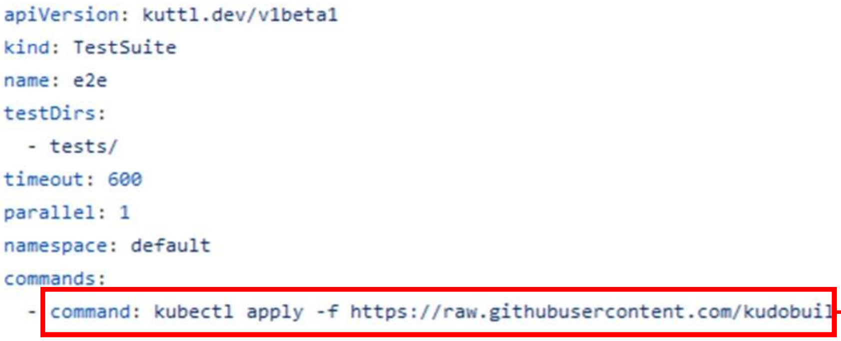
Implement KUTTL Framework (1/2)

Replace the bash scripts for the KUbernetes Test TooL (https://kuttl.dev/) framework for validation of Test Cases (Free5Gc & OAI)

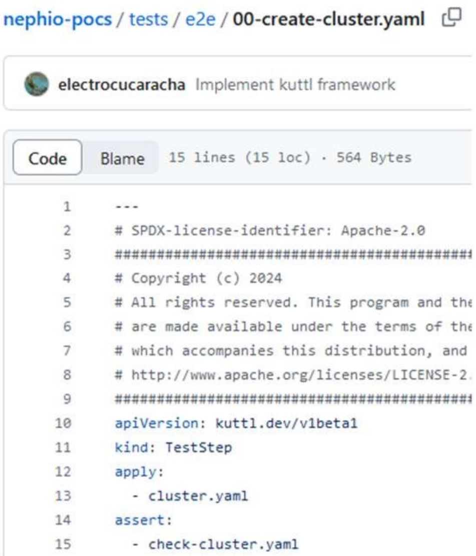
- https://github.com/nephio-project/test-infra/tree/main/e2e/tests/free5gc
- https://github.com/nephio-project/test-infra/tree/main/e2e/tests/oai

Pros	Cons
KRM testing approach	Learning curve
Easy to declare the expected state	KREW plugin requirement
Reduce instructions and complexity	https://github.com/nephio-project/nephio/issues/593

Implement KUTTL Framework (2/2)



```yaml
apiVersion: kuttl.dev/v1beta1
kind: TestSuite
name: e2e
testDirs:
  - tests/
timeout: 600
parallel: 1
namespace: default
commands:
  - command: kubectl apply -f https://raw.githubusercontent.com/kudobuil
```

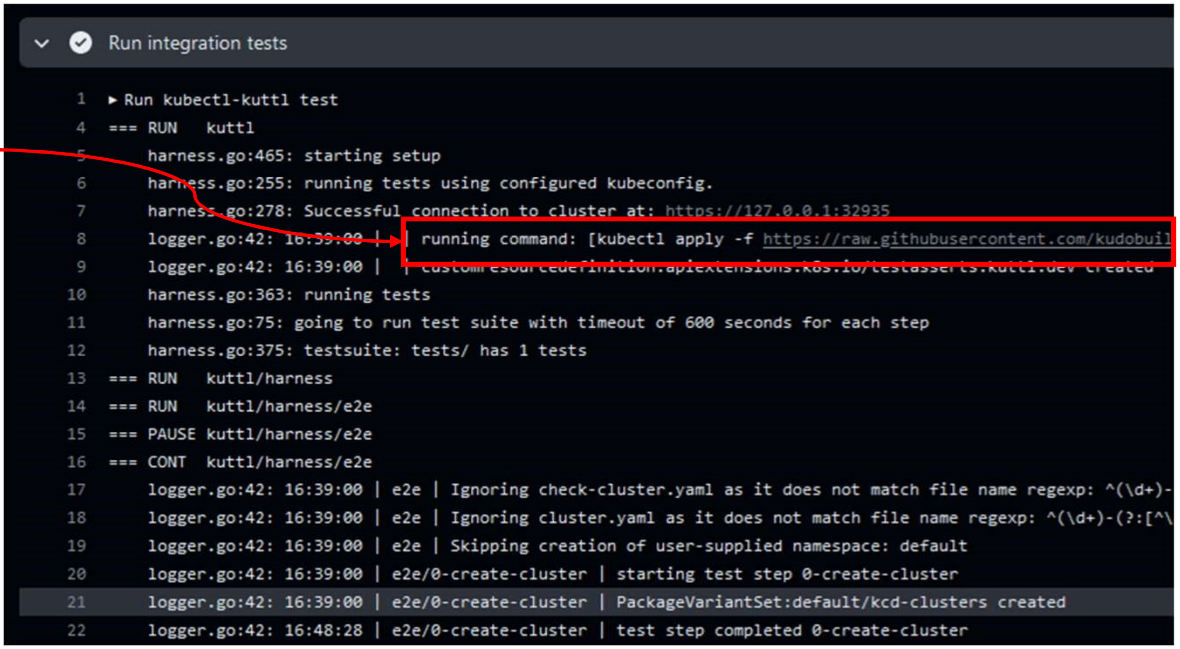


nephio-pocs / tests / e2e / 00-create-cluster.yaml

electrocucaracha Implement kuttl framework

15 lines (15 loc) · 564 Bytes

```yaml
---
# SPDX-license-identifier: Apache-2.0
###########################################
# Copyright (c) 2024
# All rights reserved. This program and the
# are made available under the terms of the
# which accompanies this distribution, and
# http://www.apache.org/licenses/LICENSE-2.
###########################################
apiVersion: kuttl.dev/v1beta1
kind: TestStep
apply:
  - cluster.yaml
assert:
  - check-cluster.yaml
```



Run integration tests

```
▸ Run kubectl-kuttl test
=== RUN   kuttl
    harness.go:465: starting setup
    harness.go:255: running tests using configured kubeconfig.
    harness.go:278: Successful connection to cluster at: https://127.0.0.1:32935
    logger.go:42: 16:39:00 | | running command: [kubectl apply -f https://raw.githubusercontent.com/kudobuil
    logger.go:42: 16:39:00 | | customresourcedefinition.apiextensions.k8s.io/testasserts.kuttl.dev created
    harness.go:363: running tests
    harness.go:75: going to run test suite with timeout of 600 seconds for each step
    harness.go:375: testsuite: tests/ has 1 tests
=== RUN   kuttl/harness
=== RUN   kuttl/harness/e2e
=== PAUSE kuttl/harness/e2e
=== CONT  kuttl/harness/e2e
    logger.go:42: 16:39:00 | e2e | Ignoring check-cluster.yaml as it does not match file name regexp: ^(\d+)-
    logger.go:42: 16:39:00 | e2e | Ignoring cluster.yaml as it does not match file name regexp: ^(\d+)-(?:[^\
    logger.go:42: 16:39:00 | e2e | Skipping creation of user-supplied namespace: default
    logger.go:42: 16:39:00 | e2e/0-create-cluster | starting test step 0-create-cluster
    logger.go:42: 16:39:00 | e2e/0-create-cluster | PackageVariantSet:default/kcd-clusters created
    logger.go:42: 16:48:28 | e2e/0-create-cluster | test step completed 0-create-cluster
```

https://github.com/electrocucaracha/nephio-pocs/commit/7a26a9a712d4e4049921c7ae5b9aee9175017c7d

Use Doc detective tool

Docs as Tests (https://www.docsastests.com/) seems to offer a new approach to keep documentation synchronized with the changes on the project. The Doc Detective (https://doc-detective.com/) tool may provide a simple and easy solution to be implemented.

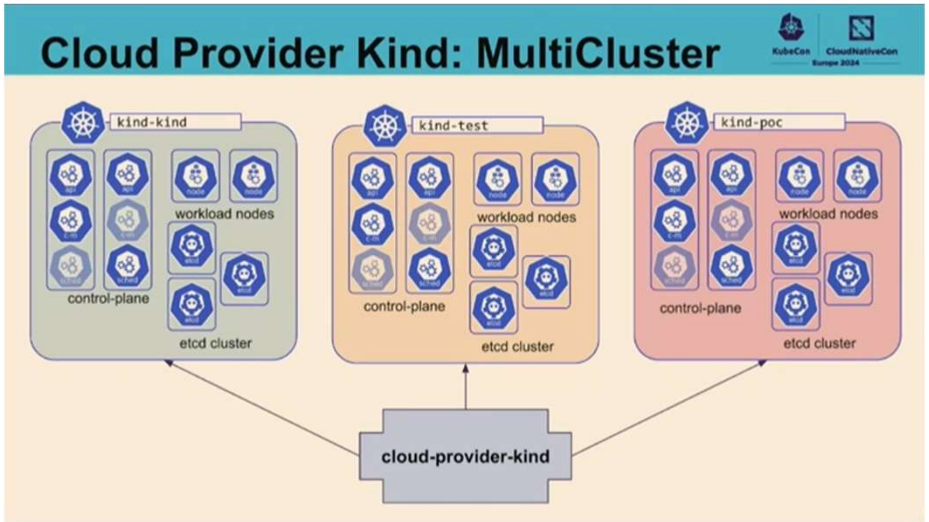
Pros	Cons
Easy way to keep docs and tests in sync	Learning curve
Non-intrusive tool	Requires further investigation (requirements and License)

Replace MetalLB for KinD cloud provider

LF NETWORKING
LFN Developer & Testing Forum

Kubernetes Cloud Provider for KIND (https://sigs.k8s.io/cloud-provider-kind) mimics the functionality offered by CSPs. This project could benefit the Nephio Sandbox.

Pros	Cons
Reduces components on the Nephio management cluster	Early stages of the tool
Simulates a more realistic scenario	Dynamic IP allocation configured on Docker default_address_pools



Cloud Provider Kind: MultiCluster

https://github.com/electrocucaracha/nephio-pocs/commit/7643a4faaeb1c7b0db77440e7f4519fe9115e320

Use Ephemeral Docker image registry

ttl.sh

We can use a Ephemeral Docker image registry in the CI (http://ttl.sh/) to replicate CI environments locally

Pros	Cons
Sharing temporal Docker images	External service (increase network traffic or delays)
Delegates the responsibility to manage temporal build images	Additional dependencies

More ideas?