

# Functest K8s v1.23

README.md (source from Functest-Kubernetes)

## Functest

Network virtualization has dramatically modified our architectures which asks for more automation and powerful testing tools like Functest, a collection of state-of-the-art virtual infrastructure test suites, including automatic VNF testing (cf. [\[1\]](#)).

In context of OPNFV, Functest verifies any kind of OpenStack and Kubernetes deployments including production environments. It conforms to upstream rules and integrates smoothly lots of the test cases available in the opensource market. It includes about 3000+ functional tests and 3 hours upstream API and dataplane benchmarks. It's completed by Virtual Network Function deployments and testing (vIMS, vRouter and vEPC) to ensure that the platforms meet Network Functions Virtualization requirements. Raspberry PI is also supported to verify datacenters as the lowest cost (50 euros hardware and software included).

Functest releases	Kubernetes releases
v1.22	v1.22
<b>v1.23</b>	<b>v1.23</b>
v1.24	v1.24
v1.25	v1.25
v1.26	v1.26
Master	v1.27.0-alpha.0 (rolling)

## Prepare your environment

cat env

```
DEPLOY_SCENARIO=k8s-XXX
```

## Run healthcheck suite

```
sudo docker run --env-file env \  
-v $(pwd)/config:/root/.kube/config \  
opnfv/functest-kubernetes-healthcheck:v1.23
```

TEST CASE	PROJECT	TIER	DURATION	RESULT
k8s_quick	functest	healthcheck	00:18	PASS
k8s_smoke	functest	healthcheck	01:14	PASS

## Run smoke suite

```
sudo docker run --env-file env \  
-v $(pwd)/config:/root/.kube/config \  
opnfv/functest-kubernetes-smoke:v1.23
```

TEST CASE	PROJECT	TIER	DURATION	RESULT
xrally_kubernetes	functest	smoke	13:00	PASS
k8s_conformance	functest	smoke	16:31	PASS
k8s_conformance_serial	functest	smoke	15:34	PASS
sig_api_machinery	functest	smoke	09:01	PASS
sig_api_machinery_serial	functest	smoke	01:24	PASS
sig_apps	functest	smoke	03:45	PASS
sig_apps_serial	functest	smoke	00:31	PASS
sig_auth	functest	smoke	09:04	PASS
sig_cluster_lifecycle	functest	smoke	00:26	PASS
sig_instrumentation	functest	smoke	00:03	PASS
sig_network	functest	smoke	05:43	PASS
sig_node	functest	smoke	28:03	PASS
sig_scheduling_serial	functest	smoke	08:03	PASS
sig_storage	functest	smoke	09:17	PASS
sig_storage_serial	functest	smoke	02:40	PASS

## Run security suite

```
sudo docker run --env-file env \
-v $(pwd)/config:/root/.kube/config \
opnfv/functest-kubernetes-security:v1.23
```

TEST CASE	PROJECT	TIER	DURATION	RESULT
kube_hunter	functest	security	00:19	PASS
kube_bench_master	functest	security	00:02	PASS
kube_bench_node	functest	security	00:01	PASS

## Run benchmarking suite

```
sudo docker run --env-file env \
-v $(pwd)/config:/root/.kube/config \
opnfv/functest-kubernetes-benchmarking:v1.23
```

TEST CASE	PROJECT	TIER	DURATION
xrally_kubernetes_full	functest	benchmarking	33:07
netperf	functest	benchmarking	40:16

## Run cnf suite

```
sudo docker run --env-file env \
-v $(pwd)/config:/root/.kube/config \
opnfv/functest-kubernetes-cnf:v1.23
```

TEST CASE	PROJECT	TIER	DURATION	RESULT
k8s_vims	functest	cnf	09:06	PASS
helm_vims	functest	cnf	08:54	PASS
cnf_testsuite	functest	cnf	16:47	PASS

## Use on air gap environments (no access to Internet)

To test a Kubernetes without access to Internet, repository mirrors needs to be provided.

Currently, all tests supports this feature except cnf conformance.

There's two ways for providing the repository mirrors:

- Give an environment variable (MIRROR\_REPO) which gives a repository with all needed images.
- Gives an environment variable per needed repo:
  - DOCKERHUB\_REPO for DockerHub repository (docker.io)
  - GCR\_REPO for Google Cloud repository (gcr.io)
  - K8S\_GCR\_REPO for Kubernetes repository (k8s.gcr.io)
  - QUAY\_REPO for Quay repository (quay.io)

All needed images are given in [functest\\_kubernetes/ci/images.txt](#)

For e2e tests, docker.io is hardcoded. It does mean that you'll have to set up a mirror on docker. An example on how to set it up on docker daemon is provided here: [daemon-configuration-file](#)