

EDEN

MOVED TO <https://wiki.lfedge.org/display/EVE/EVE>

v.0.1 — WIP

Overview

The Eden project is about:-

- Build & deploy EVE with ADAM and other runtime supplements in Docker containers and control this deployment
- Administer existing configurations (now only one or controller command)
- Upgrade EVE images on any amount of devices
- Handle test suits and configuration of tests
- Run runtime test suits and provide comprehensive logs
 - Test 1/10 : Test name 1 : Passed
 - Test 2/10 : Test name 2: Failed log:-
 - <error>
- Discover EVEs and get comprehensive configurations from ADAM
- Monitor existing EVE/ADAM for critical errors
- View EVE and ADAM logs and filter logs-
- Deploy containers/EVE VM's to any amount of EVEs

Questions:-

- Does user change config manually — yes validate the config overall — syntax — optional (default on)-
- Now we don't have options to substitute modules — no package manager for EVE

Building EVE

builds are done in a separate place. Eden can work only with pre built binaries-

Deploy EVE/ADAM

Now EDEN supports only one ADAM/ one EVE

- eden-config + eden-setup + eden-start — creates config, downloads base docker images + repack docker image for HTTP access from EVE-
- eden-status-
- I should be able to set a command for taking eve from my git repo or local folder, build it and deploy-
- If I change the config it should be tested against formal requirements before any deploy/change is done

Work with existing EVE/ADAM

I wanna be able to pass explicitly ADAM parameters and use ADAM + EVEs that are connected to this ADAM to

- start
- stop
- update eve image
- update eve certificates
- reboot
- create/delete/update/list networks
- create/delete/update/list resources (e.g. external mounts)
- run tests and get their status in a comprehensive way
- get status of the EVE/ADAM components
- list EVE workload components and their status
- get logs from any EVE/ADAM
- get logs from EVE/ADAM workloads

Running tests

Test bundles. — A list of go tests-

I should be able to list test bundles

benchmark_1.lst

— test1-

— test2-

inside /tests/test1/ ././

I should be able to create test bundle by listing individual items

I should be able to get a comprehensive description of a test bundle

I should be able to run a test bundle and on each stage be sure what happens with the test. Get what item am I running and it's stage.

I should be able to turn off (and it should be default option) any output except SUCCESS/PASSED for individual items

I should be able at any time get a sequential number and the name of the test in the itemlist

I should get back the error of the test once it occurs.

I should have an option to stop the test on the first occurred error or continue. In that case I would like to see

Test 1/10: Test name 1 — PASSED

Test 2/10: Test name 2 — FAILED

<Error log>

Test 3/10: Test name 3 — PASSED

Proposed commands:

Config

A config is a configuration of eden. It contains EDEN setup and handles the default config that is used in other commands

- *eden config generate default*
- *eden config add*
- *eden config list*
- *eden config delete*
- *eden config set context*

Eden

Eden itself as a bundle. When Eden is started or stopped all services within eden are started and stopped.

Status returns the list of all eden services from the current config.

- *eden start/stop/restart/status*

Eden-adam

Work with adams in the eden config

- *eden adam list*
- *eden adam start/stop/restart/status*
- *eden adam get config*
- *eden adam update config*
- *eden adam logs*

Eden-eve

Work with eve in the eden config

- *eden eve list*
- *eden eve start/stop/restart/status*
- *eden eve eveimage update*
- *eden eve eveimage remove*
- *eden eve update config*
- *eden eve logs*
- *eden eve get config*
- *eden eve attach*
- *eden eve exec*

Eden-tests

Perform tests within current eden config. Works with test bundles that are a list of go tests executed sequentially

- *eden tests list*

- eden tests add
- eden tests delete
- eden tests start
- eden tests stop
- eden tests logs

Eden pods

~~Work with containers within eves from eden config~~

- eden pod list
- eden pod add
- eden pod run
- eden pod delete
- eden pod start/stop/restart/status
- eden pod logs
- eden pod attach
- eden pod exec