LF NETWORKING

LFN Developer & Testing Forum

LF NETWORKING

LFN Developer & Testing Forum

Custom script execution in CDS and what we can do with it.

Marek Szwałkiewicz – T-Mobile PL

Anti-Trust Policy Notice

LFN NETWORKING
LFN Developer & Testing Forum

- Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

- Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at http://www.linuxfoundation.org/antitrustpolicy. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Updegrove LLP, which provides legal counsel to the Linux Foundation.

CDS is good, this talk is not a critique

LFN Developer & Testing Forum

- They create(d) an amazing tool
- A lot of good decisions was made
- I think too much about Python
- Good things can be improved

181	Ezhilarasi
154	Brinda Santh
144	Arundathi Patil
130	Alexis de Talhouët
121	Muthuramalingam, Brinda Santh(bs2796)
84	Muthuramalingam, Brinda Santh
65	Oleg Mitsura
65	Singal, Kapil (ks220y)
61	ShaabanEltanany
58	Timoney, Dan (dt5972)
53	Ahmedeldeeb50
50	Jozsef Csongvai
49	Dan Timoney
41	Sarah Abouzainah
39	Swapnali Shadanan Pode
34	JakobKrieg
27	Julien Fontaine
23	Steve Siani
22	shaaban Altanany
21	Lukasz Rajewski
18	AhmedEldeeb50
18	Eltanany Shaaban
17	Serge Simard
17	Steve Alphonse Siani
16	Ahmed Abbas
15	Eliezio Oliveira
15	ezhil
15	ottera
15	Abdelmuhaimen Seaudi

A bit of backstory

- Business driven day2 configuration
- Customization of orchestration logic for operator's needs
- Integration of vendors' specific interfaces
- Use case driven orchestration logic

What options do we have?

Available “executors” in CDS:

- kotlin?
- py-executor
- command-executor
  - Ansible
  - Python
- cli-executor
- restconf-executor
- restful-executor
- netconf-executor
- (…)

What is happening in command-executor

LFN Developer & Testing Forum

```
251             # SR7/SR10 compatibility hack
252             # check if the path for the request.command does not contain UUID, then add it after cba_name/cba_version path.
253             updated_request_command = request.command
254             if self.blueprint_name_version in updated_request_command and self.blueprint_name_version_uuid not in updated_request_command:
255                 updated_request_command = updated_request_command.replace(self.blueprint_name_version, self.blueprint_name_version_uuid)
256
257             if "ansible-playbook" in updated_request_command:
258                 cmd = cmd + "; " + updated_request_command + " -e 'ansible_python_interpreter=" + self.blueprint_dir + "/bin/python'"
259             else:
260                 cmd = cmd + "; " + updated_request_command + properties
261
262             ### extract the original header request into sys-env variables
263             # OriginatorID
264             originator_id = request.originatorId
265             # CorrelationID
266             correlation_id = request.correlationId
267             request_id_map = {'CDS_REQUEST_ID':self.request_id, 'CDS_SUBREQUEST_ID':self.sub_request_id, 'CDS_ORIGINATOR_ID': originator_id, 'CDS_CORRELATION_ID':
268             updated_env =  { **os.environ, **request_id_map }
269             # Prepare PATH and VENV_HOME
270             updated_env['PATH'] = self.blueprint_dir + "/bin/:" + os.environ['PATH']
271             updated_env['VIRTUAL_ENV'] = self.blueprint_dir
272             self.logger.info("Running blueprint {} with timeout: {}".format(self.blueprint_name_version_uuid, self.execution_timeout), extra=self.extra)
273             with tempfile.TemporaryFile(mode="w+") as tmp:
274                 try:
275                     completed_subprocess = subprocess.run(cmd, stdout=tmp, stderr=subprocess.STDOUT, shell=True,
276                                                           env=updated_env, timeout=self.execution_timeout)
277                 except TimeoutExpired as timeout_ex:
278                     self.prometheus_counter.labels(self.PROMETHEUS_METRICS_EXEC_COMMAND_LABEL, self.blueprint_name, self.blueprint_version, request.command).inc()
279                     timeout_err_msg = "Running command {} failed due to timeout of {} seconds.".format(self.blueprint_name_version_uuid, self.execution_timeout)
280                     self.logger.error(timeout_err_msg, extra=self.extra)
281                     # In the time-out case, we will never get CBA's script err msg string.
282                     utils.parse_cmd_exec_output(outputfile=tmp, logger=self.logger, payload_result=result, err_msg_result=script_err_msg, results_log=results_log,
283                     return utils.build_ret_data(False, results_log=results_log, error=timeout_err_msg)
284                 utils.parse_cmd_exec_output(outputfile=tmp, logger=self.logger, payload_result=result, err_msg_result=script_err_msg, results_log=results_log, extr
285                 rc = completed_subprocess.returncode
286         except Exception as e:
287             self.prometheus_counter.labels(self.PROMETHEUS_METRICS_EXEC_COMMAND_LABEL, self.blueprint_name, self.blueprint_version, request.command).inc()
288             err_msg = "{} - Failed to execute command. Error: {}".format(self.blueprint_name_version_uuid, e)
289             result.update(utils.build_ret_data(False, results_log=results_log, error=err_msg))
290             return result
291
292         # Since return code is only used to check if it's zero (success), we can just return success flag instead.
```

What is happening in py-executor


```python
def instance_for_input(config: ScriptExecutorConfiguration, input: ExecutionServiceInput):
    blueprint_name = input.actionIdentifiers.blueprintName
    blueprint_version = input.actionIdentifiers.blueprintVersion
    action_name = input.actionIdentifiers.actionName
    # Get Blueprint python script location
    script_location = blueprint_location(config, input) + '/' + 'Scripts/python/__init__.py'
    logger.info(script_location)

    # Create Dynamic Module Name
    module_name = blueprint_name + '-' + blueprint_version
    spec = importlib.util.spec_from_file_location(module_name, script_location)
    logger.info(spec)
    dynamic_module = importlib.util.module_from_spec(spec)
    # Add blueprint modules
    sys.modules[spec.name] = dynamic_module
    spec.loader.exec_module(dynamic_module)      Brinda Santh, 3 years ago • Add bi-directional GRPC python executor. …
    script_clazz = getattr(dynamic_module, action_name)
    return script_clazz()
```

What can go wrong?

- Ups I deleted wrong log
- I messed in nfs share
- Someone was a bad person in the past
  - Bad dependencies
  - Bad dependencies of dependencies
  - Attacks on build systems (build integrity)
  - Attacks on repositories (source integrity)
  - Mom, someone is mining on my container

Lets solve it

- Let's use more docker!
- We shouldn’t write yet more code
- Maybe we could share with other ONAP components?
- AWS Lambda was a good idea

Lets solve it

Lets solve it

- Fission is
  - Cloud native
  - Fast (100ms cold start)
  - Developed and bugfixed by others

```
$ fission function create --name headers --env python --code headers.py
```

LF NETWORKING

LFN Developer & Testing Forum