TeraFlow SDN by ETSI

TFS HACKFEST 2

SCHALKE TeMeto

21/06/2023

Challenges

Create your local testbed

- Deploy TFS
- Deploy Scenario in ContainerLab

Onboard the Devices from the ContainerLab Scenario in TFS

- Some descriptors are provided as reference (~/tfs-ctrl/hackfest/containerlab)

Configure a Service with Static Routing in the devices

- Some descriptors are provided as reference (~/tfs-ctrl/hackfest/containerlab)

Monitor the traffic in the device ports through Grafana

OCNOS onboarding

UUID: ad26e7c5-5c32-584e-830b-d93e565d2e70
Name: OCNOS
Type: packet-router
Status: UNDEFINED
Drivers:

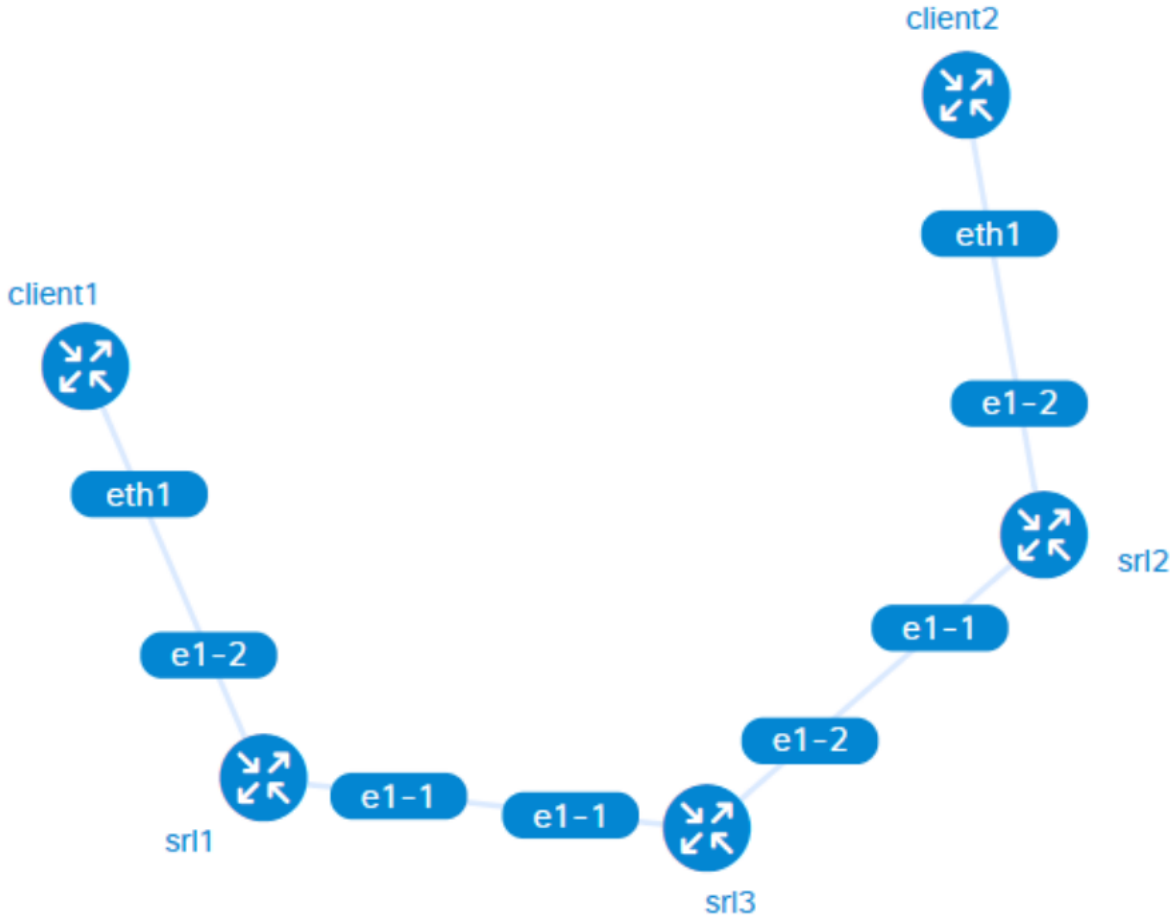
- OPENCONFIG

Endpoint UUID	Name

Configurations:

Key	Value
_connect/address	• 10.95.86.21
_connect/port	• 830
_connect/settings	• **allow_agent:** False • **commit_per_rule:** True • **device_params:** {'name': 'huaweiyang'} • **force_running:** False • **hostkey_verify:** False • **look_for_keys:** False • **manager_params:** {'timeout': 120} • **message_renderer:** pyangbind • **password:** ocnos • **username:** ocnos • **vendor:** OCNOS

Scenario with additional routers



VLAN implementation

TeraFlow SDN

```
if_enabled       = bool(resource_value.get('enabled'            , True))    # True/False
sif_enabled      = bool(resource_value.get('sub_if_enabled'     , True))    # True/False
sif_ipv4_enabled = bool(resource_value.get('sub_if_ipv4_enabled', True))    # True/False
sif_ipv4_address = str (resource_value['sub_if_ipv4_address'          ])    # 172.16.0.1
sif_ipv4_prefix  = int (resource_value['sub_if_ipv4_prefix'           ])    # 24
sif_vlan         = int (resource_value['sub_if_vlan'                  ])    # 111

str_path = '/interfaces/interface[name={:s}]'.format(if_name)
str_data = json.dumps({
    'name': if_name,
    'config': {'name': if_name, 'enabled': if_enabled},
    'subinterfaces': {
        'subinterface': {
            'index': sif_index,
            'config': {'index': sif_index, 'enabled': sif_enabled},
            'ipv4': {
                'config': {'enabled': sif_ipv4_enabled},
                'addresses': {
                    'address': {
                        'ip': sif_ipv4_address,
                        'config': {'ip': sif_ipv4_address, 'prefix_length': sif_ipv4_prefix},
                    }
                }
            }
            'vlan' : {
                'match' : {
                    'single-tagged' : {
                        'config' : {
                            'vlan-id' : vlan
                        }
                    }
                }
            }
        }
    }
})
return str_path, str_data
```

```
def _interface(if_name, sif_index, ipv4_address, ipv4_prefix, vlan enabled) -> Tuple[str, Dict]:
    str_path = '/interface[{:s}]'.format(if_name)
    str_data = {'name': if_name, 'enabled': enabled, 'sub_if_index': sif_index,
                'sub_if_enabled': enabled, 'sub_if_ipv4_enabled': enabled,
                'sub_if_ipv4_address': ipv4_address, 'sub_if_ipv4_prefix': ipv4_prefix, 'sub_if_vlan': vlan}
    return str_path, str_data
```

```
class EndpointComposer:
    def __init__(self, endpoint_uuid : str) -> None:
        self.uuid = endpoint_uuid
        self.objekt : Optional[EndPoint] = None
        self.sub_interface_index = 0
        self.ipv4_address = None
        self.ipv4_prefix = None
        self.vlan        = None

    def configure(self, endpoint_obj : EndPoint, settings : Optional[TreeNode]) -> None:
        self.objekt = endpoint_obj
        if settings is None: return
        json_settings : Dict = settings.value
        self.ipv4_address = json_settings['ipv4_address']
        self.ipv4_prefix = json_settings['ipv4_prefix']
        self.sub_interface_index = json_settings['sub_interface_index']
        self.vlan = json_settings['vlan']

def get_config_rules(self, network_instance_name : str, delete : bool = False) -> List[Dict]:
    json_config_rule = json_config_rule_delete if delete else json_config_rule_set
    return [
        json_config_rule(*_interface(
            self.objekt.name, self.sub_interface_index, self.ipv4_address, self.ipv4_prefix,self.vlan, True
        )),
        json_config_rule(*_network_instance_interface(
            network_instance_name, self.objekt.name, self.sub_interface_index
        )),
    ]
```

TeraFlow SDN by ETSI

Thank You!