Monitoring VNFs Using D-MONA

Onboarding D-MONA

Cisco Elastic Services Controller supports Distributed Monitoring and Actions (D-MONA) for effective monitoring of the VNFs. D-MONA is a standalone monitoring application. For more information, see Monitoring VNFs Using D-MONA in the Cisco Elastic Services Controller User Guide.

To onboard D-MONA, you must fulfill the prerequisites and prepare the deployment data model:

Prerequisites

- Ensure connectivity between ESC and D-MONA.
- Ensure connectivity between D-MONA and the deployed VNFs.

Note Monitoring of D-MONA by another D-MONA is not supported.

For information on deploying D-MONA, see Deploying D-MONA, on page 1.

Deploying D-MONA

From ESC 5.3 or later, 1:1 mapping is not required. It supports explicit D-MONA deployment.

- In this scenarios, multiple D-MONA Instances can be deployed.
- VNFs can be deployed under, or migrated to specified monitoring agent.

For more information on deploying the VNFs with explicit D-MONA mapping, see the Deploying VNFs with Explicit D-MONA Mapping chapter in the Cisco Elastic Services Controller User Guide.

For using D-MONA in your infrastructure, you must:

1. Deploy the D-MONA with the monitoring infrastructure.
2. Deploy the VNFs using the D-MONA for monitoring.

After deployment, D-MONA is monitored by the local MONA running on the ESC VM.

The following example shows the D-MONA VNFD:

```
tosca_definitions_version: tosca_simple_yaml_1_3
description: D-MONA VNFD (SOL001 v0.10.0)

imports:
  - cisco_nfv_sol001_types.yaml
  - etsi_nfv_sol001_vnfd_0_10_0_types.yaml

metadata:
  template_name: D-MONA
  template_author: Cisco Systems
  template_version: '1.0'

dsl_definitions:
  descriptor_id: &descriptor_id f5b37b47-d9bd-4605-afb0-30c0d659a3c2
  provider: &provider cisco
  product_name: &product_name D-MONA
  software_version: &software_version '1.0'
  descriptor_version: &descriptor_version '1.0'
  flavour_id: &flavour_id default
  flavour_description: &flavour_description 'Default VNF Deployment Flavour'
  vnfm: &vnfm '9:Cisco Elastic Services Controller:v04.04.01'
```

For information on deploying VNFs using D-MONA, see Using D-MONA for a Deployed VNF, on page 4.

Table 1: Input Parameters for D-MONA Deployment

Parameter	Description
SW_IMAGE_NAME	The name of ESC image
DMONA_CERT	The HTTPS certificate
ADMIN_PASSWORD	The admin user password
SECURITY_BASIC_ENABLED	A flag that indicates whether basic security is enabled or not
SECURITY_USER_NAME	A security user to communicate with ESCManager
SECURITY_USER_PASSWORD	A security user's password used to communicate with ESCManager

KPI data:

- property_list
  - name—protocol

- value—https
- name—port
- value—8443
- name—path
- value—mona/v1/health/status
- name—application_startup_timevalue—true

Config data parameters:

- user-data.txt

  admin_password—value defined for ADMIN_PASSWORD in input parameter
- application—dmona.template
  - monitoring.agent—true
  - security_basic_enabled—value defined for SECURITY_BASIC_ENABLED in input parameter
  - security_user_name—value defined for SECURITY_USER_NAME in input parameter
  - security_user_password—value defined for SECURITY_USER_PASSWORD in input parameter
  - monitoring.agent.vim.mapping—false

Example payload:

```
config_data:
    '--user-data':
        file: ../Files/Scripts/user-data.txt
        variables:
             admin_password: { get_input: ADMIN_PASSWORD }
    '/opt/cisco/esc/mona/dmona.crt':
        data: { get_input: DMONA_CERT }
    '/opt/cisco/esc/mona/config/application-dmona.properties':
        file: ../Files/Scripts/application-dmona.template
        variables:
             monitoring.agent: true
             security_basic_enabled: { get_input: SECURITY_BASIC_ENABLED }
             security_user_name: { get_input: SECURITY_USER_NAME }
            security_user_password: { get_input: SECURITY_USER_PASSWORD }
            monitoring.agent.vim.mapping: false
```

The following table lists the D-MONA VM flavors for large scale deployments:

Deployment	Number of VMs	Virtual CPU per VM	Virtual Memory (GB) per VM	Virtual Hard Disk (GB)per VM	Number of total VMs Supported
D-MONA	1	4	8	40	1500

Configuring D-MONA

While configuring D-MONA, you can view two types of runtime behavior; one from a typical ESC deployment, and the other one with capabilities provided by D-MONA.

D-MONA Day Zero Configuration

The D-MONA runtime behavior is controlled by the day 0 configuration provided to the VM at the time of deployment.

The following example shows D-MONA SSH access and D-MONA ESC certificate configuration:

```
            config_data:
              '--user-data':
                file: file:///opt/cisco/esc/esc-config/dmona/user-data.template
                variables:
                  # This is the SHA-512 hashed password for 'C1sco@123'
                  vm_credentials:
$6$rounds=4096$6YN5.SHEdfa6v$t6tkvtIrEZv9xpFlLIKkkU2CBq6G2rtObztMqui4Y7uRUBDU62T0NIeDpMn4/TPMsbiBL8CHjdjZaj/5HlwIo/

              '/opt/cisco/esc/mona/dmona.crt':
                data: { get_input: DMONA_CERT }
              '/opt/cisco/esc/mona/config/application-dmona.properties':
                file: file:///opt/cisco/esc/esc-config/dmona/application-dmona.template
```

The `vm_credentials` passes the encrypted password to admin for SSH access to D-MONA.

For monitoring using D-MONA, see Monitoring Using D-MONA, on page 6.

Using D-MONA for a Deployed VNF

For deploying the VNFs using D-MONA for monitoring, you must have the D-MONA with the `monitoring.agent.vim.mapping day-0` variable set to false. When ESC detects D-MONA, monitoring of the VNF is assigned to that D-MONA, otherwise the local MONA handles the monitoring.

Specifying D-MONA Monitoring Agent through ETSI ESC Interface

Use the following to specify the monitoring agent:

1. Only specify the monitoring agent (via additionalParams), or you can have the monitoring agent under KPI section of VNFD.

   Here, the specified agent is used to populate the deployment model processed by ESC.

   NFVO or EM sends the POST request.

   Method Type:

   POST

   VNFM Endpoint:

   `vnflcm/v2/vnf_instances/$vnf_instance_id/instantiate`

Example:

```
InstantiateVnfRequest with only the monitoring agent specified (additionalParams)

############################
# Instantiate VNF Request #
############################

#POST https://localhost:8251/vnflcm/v2/vnf_instances/$vnf_instance_id/instantiate
{
    "flavorId": "default",
    "instantiationLevelId": "default",
    "vimConnectionInfo": {
        "default_openstack_vim": {
            "accessInfo": {
                "password": "VIM-password",
                "project": "Project_001",
                "projectDomain": "default",
                "region": "regionOne",
                "userDomain": "VIM-user-uuid",
                "username": "VIM-user"
            },
               "interfaceInfo": {
                "endpoint": "http://openstack_vim:5000/v3/auth"
            },
            "vimId": "VIM-001",
            "vimType": "OPENSTACK_V3"
        }
    },
    "extVirtualLinks": [
        {
            "id": "Network0",
            "extCps": [
                {
                    "cpConfig": {
                      "cp1": {
                            "cpProtocolData": [
                                {
                                    "ipOverEthernet": {
                                        "ipAddresses": [
                                            {
                                                "subnetId":
"654c5793-c74b-4e78-8bd5-2162ec3f9f3e",
                                                "type": "IPV4"
                                            }
                                        ]
                                    },
                                    "layerProtocol": "IP_OVER_ETHERNET"
                                }
                            ]
                        }
                    ],
                    "cpdId": "VDU_1:port_1"
                }
            ],
            "resourceId": "3ecaeb96-f2f5-4eed-b51f-8a69e80748f3",
            "resourceProviderId": "3ecaeb96-f2f5-4eed-b51f-8a69e80748f3",
            "vimConnectionId": "string"
        }
    ],
    "additionalParams": {
        "CF1_SLOT_CARD_NUMBER": "1",
        "CF2_SLOT_CARD_NUMBER": "2",
        "CF_CARD_TYPE_NUM": "0x40010100",
```

```
            "CF_DOMAIN_NAME": "cisco.com",
            "CF_NAME_SERVER": "171.70.168.183",
            "CF_STAROS_CONFIG_URL": "../Files/Scripts/control-function/staros_config.txt",
            "CF_STAROS_PARAM_URL": "../Files/Scripts/control-function/staros_param_cf.cfg",

            "CF_VIP_ADDR": "172.77.11.6",
            "CHASSIS_KEY": "164c03a0-eebb-44a8-87fa-20c791c0aa6d",
            "SF1_SLOT_CARD_NUMBER": "3",
            "SF2_SLOT_CARD_NUMBER": "4",
            "SF_CARD_TYPE_NUM": "0x42030100",
            "SF_STAROS_PARAM_URL": "../Files/Scripts/session-function/staros_param_sf.cfg",

            "VIM_NETWORK_DI_INTERNAL1": "etsi-vpc-di-internal1",
            "VIM_NETWORK_DI_INTERNAL2": "etsi-vpc-di-internal2",
            "VIM_NETWORK_MANAGEMENT": "DualStack-KPI-M-Test-Net",
            "VIM_NETWORK_ORCHESTRATION": "esc-net",
            "VIM_NETWORK_SERVICE1": "etsi-vpc-service1",
            "VIM_NETWORK_SERVICE2": "etsi-vpc-service2",
            "VNFM_PROXY_ADDRS": "172.77.12.106,172.77.12.104,172.77.12.105",
            "VNFM_MONITORING_AGENT": "dmonaName://dm1-agent"
        }
    }
```

The single agent specified in the API request is mapped to the variable wherever it is used in the VNFD and is converted to the appropriate data model.

Monitoring Using D-MONA

To monitor the VNFs using D-MONA, you must deploy the ETSI VNFD D-MONA and then deploy the ETSI VNFD monitored by D-MONA.

The D-MONA parameters are defined within the VNFD, or provided as additionalparams in the instantiate D-MONA VNF payload.

An ETSI compliant VNFD is used for the deployment of D-MONA.

The input parameters, KPI data, and config paramters are required for instantiation of D-MONA deployment.

The input parameters are either defined within the VNFD or provided as additionalParams section of instantiate D-MONA VNF payload.

```
"VNFM_MONITORING_AGENT": "dmonaName://<dmona_instance__name>"
```

Resetting the Monitoring Rules for D-MONA

ESC can now detect rebooting of the D-MONA application by monitoring the startup time.

The local MONA monitors D-MONA by performing a HTTP(S) call to the D-MONA health API and keeps track of the last known startup time of the polled D-MONA process. Upon successful request (status code = 200), local MONA compares the last known startup time with the returned startup time from the polled application.

To enable the startup time check, you must set `application_startup_time` to true in KPI section of VNFD yaml.

However, if the `application_startup_time` is not present or set to false, then DMONA reboot check is disabled. You must set this property for D-MONA reboot.

Note The application startup time is not backward compatible. It is available from ESC release 5.3 onwards.

Following is a sample KPI section of D-MONA VNFD:

```
 VM_ALIVE-1:
                event_name: 'VM_ALIVE'
                metric_value: 1
                metric_cond: 'GT'
                metric_type: 'UINT32'
                metric_occurrences_true: 1
                metric_occurrences_false: 30
                metric_collector:
                  type: 'HTTPGET'
                  nicid: 0
                  address_id: 0
                  poll_frequency: 3
                  polling_unit: 'seconds'
                  continuous_alarm: false
                  property_list:
                    - name: protocol
                      value: https
                    - name: port
                      value: 8443
                    - name: path
                      value: mona/v1/health/status
                    - name: application_startup_time # Set to true to enable start time
check
                      value: true
```