Monitoring VNF Using D-MONA

The ESC Monitoring and Action (MONA) monitors VNFs that are deployed by ESC. To maintain accuracy, it executes actions, such as ping, custom_scripts, and so on at specific intervals.

Onboarding D-MONA

The following prerequisites must be fulfilled before deploying D-MONA:

Prerequisites

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

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

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

Deploying D-MONA

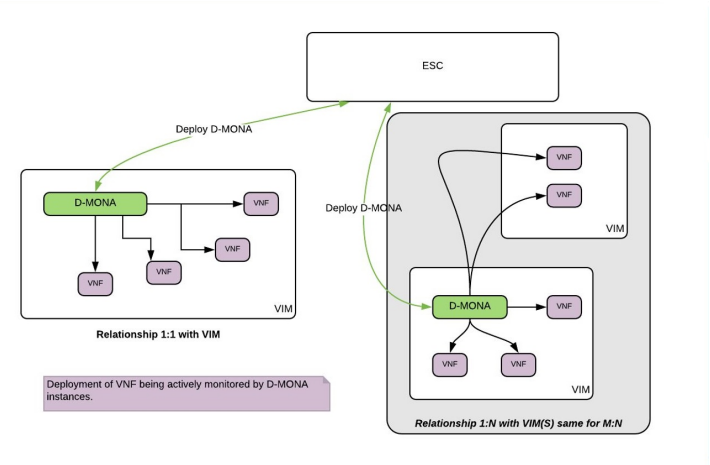
From ESC 5.3 or later, 1:1 mapping is not required. It supports explicit D-MONA deployment.

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

For 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 the monitoring.

Figure 1: D-MONA Deployment Types



If you are not using D-MONA for monitoring, see Monitoring Virtual Network Functions section.

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 2 types of runtime behavior, one where you can view the full behavior expected from a typical ESC deployment, and the other one depicts the capabilities provided by D-MONA.

The D-MONA runtime behavior is controlled by the day-0 configuration that is provided to the VM at deployment time. For more information on day zero configuration, see the D-MONA Day Zero Configuration section.

You must provide the notification URL for HA Active/Standby and Standalone. However, for the Active/Active HA, the URL is auto-generated or computed during the deployment.

D-MONA Day Zero Configuration

The following example shows D-MONA SSH VM access configuration:

```
<configuration>
 <dst>--user-data</dst>
 <file>file:///opt/cisco/esc/esc-config/dmona/user-data.template</file>
 <variable>
  <name>vm_credentials</name>
  <val>REPLACED_WITH_GENERATED_PWD</val>
```

```
 </variable>
</configuration>
```

The following example shows the notification URL for HA Active/Standby and Standalone:

```
<variable>
  <name>notification.url</name>
  <val>
    http(s)://xxx.xx.x.xx:xxxx/ESCManager/dmona/api/events/notif
  </val>
</variable>
```

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

The following example shows the D-MONA ESC certificate configuration:

```
<configuration>
 <dst>/opt/cisco/esc/moan/dmona.crt</dst>
 <data>$DMONA_CERT</data>
</configuration>
```

The following example shows the D-MONA application user data configuration:

```
<configuration>
<dst/opt/cisco/esc/mona/config/application-dmona.properties</dst>
 <file>file:///opt/cisco/esc/esc-config/dmona/application-dmona.template</file>
 <variable>
  <name>monitoring.agent</name>
  <val>true</val>
 </variable>
 <variable>
  <name>monitoring.agent.vim.mapping</name>
  <val>true</val>
 </variable>
 <!—Used to enable Basic Authentication for communication with the D-MONA Application.->
 <variable>
  <name>security_basic_enabled</name>
  <val>true</val>
 </variable>
 <variable>
  <name>security_user_name</name>
  <val>REPLACED_WITH_USER_NAME</val>
 </variable>
 <variable>
  <name>security_user_password</name>
  <val>REPLACED_WITH_USER_PASSWORD</val>
 </variable>

</configuration>
```

The following example shows the D-MONA day-0 template file for CSP:

Upload the D-MONA day-0 template to the `/var/tmp/` directory in all the ESC instances with proper access permission prior to deployment.

```
#cloud-config
users:
 - name: admin          # The user's login name
   gecos: admin         # The user name's real name
   groups: esc-user     # add admin to group esc-user
   passwd: $vm_credentials
                        # The hash -- not the password itself -- of the password you want
                       #           to use for this user. You can generate a safe hash via:
```

```
                        #               mkpasswd --method=SHA-512 --rounds=4096
   lock-passwd: false   # Defaults to true. Lock the password to disable password login
                        # Set to false if you want to password login
   homedir: /home/admin # Optional. Set to the local path you want to use. Defaults to
/home/<username>
   sudo: ALL=(ALL) ALL  # Defaults to none. Set to the sudo string you want to use

ssh_pwauth: True        # Defaults to False. Set to True if you want to enable password
authentication for sshd.
write_files:
# ESC Configuration
- path: /opt/cisco/esc/esc-config/esc-config.yaml
  content: |
    resources:
      mona:
        dmona: true
- path: /etc/sysconfig/network-scripts/ifcfg-eth0
  content: |
    DEVICE="eth0"
    BOOTPROTO="none"
    ONBOOT="yes"
    TYPE="Ethernet"
    USERCTL="yes"
    IPADDR="${NICID_0_IP_ADDRESS}"
    NETMASK="${NICID_0_NETMASK}"
    GATEWAY="${NICID_0_GATEWAY}"
    DEFROUTE="yes"
    NM_CONTROLLED="no"
    IPV6INIT="no"
    IPV4_FAILURE_FATAL="yes"
- path: /etc/sysconfig/network-scripts/ifcfg-eth1
  content: |
    DEVICE="eth1"
    BOOTPROTO="none"
    ONBOOT="yes"
    TYPE="Ethernet"
    USERCTL="yes"
    IPADDR="${NICID_1_IP_ADDRESS}"
    NETMASK="${NICID_1_NETMASK}"
    GATEWAY="${NICID_1_GATEWAY}"
    DEFROUTE="yes"
    NM_CONTROLLED="no"
    IPV6INIT="no"
    IPV4_FAILURE_FATAL="yes"
runcmd:
- [ cloud-init-per, once, apply_network_config, sh, -c, "systemctl restart network"]
- [ cloud-init-per, once, copy_dmona_config, sh, -c, "cp -RT /media/cdrom/opt/cisco/esc/mona/
 /opt/cisco/esc/mona/"]
- [ cloud-init-per, once, esc_service_start, sh, -c, "chkconfig esc_service on && service
esc_service start"] # You must include this line
```

Deploying VNFs with Explicit D-MONA Monitoring Agent

From ESC 5.3 onwards, ESC allows to explicitly specify the D-MONA identifier to monitor a VNF. Following are the steps to deploy VNFs with explicit VNF to D-MONA monitoring agent:

Procedure

Step 1 Deploy a D-MONA with monitoring.agent.vim.mapping property in the D-MONA day-0 config omitted or set to false.

The following example shows the day 0 config of a D-MONA datamodel where monitoring.agent.vim.mapping is set to false.

```
<configuration>
        <dst>/opt/cisco/esc/mona/config/application-dmona.properties</dst>
        <file>file:///opt/cisco/esc/esc-config/dmona/application-dmona.template</file>
        <variable>
            <name>monitoring.agent</name>
            <val>true</val>
        </variable>
        <!-- property for one to one mapping - omit or set to false for explicit VNF to
D-MONA mapping-->
        <variable>
            <name>monitoring.agent.vim.mapping</name>
            <val>false</val>
        </variable>
       <!-- property to enable basic auth in dmona. Not to be confused with basic auth for
 esc -->
        <variable>
            <name>security_basic_enabled</name>
            <val>true</val>
        </variable>
        <variable>
            <name>security_user_name</name>
            <val>REPLACE_WITH_USER_NAME</val>
        </variable>
        <variable>
            <name>security_user_password</name>
            <val>REPLACE_WITH_USER_PASSWORD</val>
        </variable>
</configuration>
```

Step 2 Deploy the VNF by specifying the monitoring_agent parameter in the KPI config of the deployment datamodel.

The tag *<monitoring_agent>* is used as an explicit identification of a distributed mona deployment which monitors the VNF. When the tag is present, ESC looks for a distributed mona deployment with that exact deployment name. The D-MONA identifier is specified in the URI by using a specific scheme to represent a previously deployed D-MONA VNF.

For example, `dmonaName://<D_MONA_DEP_NAME>` Replace `<D_MONA_DEP_NAME>` with deployment name of Distributed MONA instance.

The following example shows the KPI config of a VNF datamodel with monitoring agent specified:

```
<kpi>
    <event_name>VM_ALIVE</event_name>
    <!-- specify dmona deployment name using dmonaName:// URI format-->
    <monitoring_agent>dmonaName://D-MONA-OTTAWA</monitoring_agent>
    <metric_value>1</metric_value>
    <metric_cond>GT</metric_cond>
    <metric_type>UINT32</metric_type>
    <metric_collector>
    <type>ICMPPing</type>
    <nicid>0</nicid>
    <poll_frequency>3</poll_frequency>
    <polling_unit>seconds</polling_unit>
    <continuous_alarm>false</continuous_alarm>
```

```
    <monitoring_public_ip>true</monitoring_public_ip>
    </metric_collector>
</kpi>
```

Note ESC allows only one single monitoring agent per VNF.

Troubleshooting Monitoring Status

To determine if a VNF is monitored by a monitoring agent for D-MONA, run the following command:

```
curl -u username:pwd -H 'Accept:application/json'
http://localhost:8080/ESCManager/v0/api/monitoring/agents/config
```

The sample below shows the result:

```
{
    "a8345881-adc8-4d16-8741-9d105592c676": {
        "monitoringAgents": [
            {
                "name": "sample-dmona-10",
                "notificationUrl":
"https://172.16.235.73:8443/ESCManager/dmona/api/events/notif",
                "oneToOneMapping": false,
                "state": "ACTIVE",
                "uri": "https://172.16.235.81:8443/mona/v1/rules",
                "vimId": "OPENSTACK_VIMCONN_pf-ucs-20",
                "vnfData": [
                    {
                        "deploymentExternalId": "785e170c-55b5-4df7-929f-d34f052e4616",
                        "deploymentName": "dmona-10-vnf-121-f2b1df6d",
                        "state": "MONITORED",  <============ Monitoring state for DMONA
                        "vmGroupName": "vm1"
                    },
                    {
                        "deploymentExternalId": "2e42c8d9-51fa-4de8-a260-d3a3429be7d4",
                        "deploymentName": "dmona-10-vnf-442-faa43053",
                        "state": "MONITORED",  <============ Monitoring state for DMONA
                        "vmGroupName": "vm1"
                   }
                ]
            },
            {
                "name": "local_mona",
                "notificationUrl": "",
                "oneToOneMapping": false,
                "state": "ACTIVE",
                "uri": "http://localhost:8090/mona/v1/rules",
                "vimId": "N/A",
                "vnfData": [
                    {
                        "deploymentExternalId": "9501376e-e29e-4c99-b5fb-66ab66de45b7",
                        "deploymentName": "sample-dmona-2",
                        "state": "N/A",   <============ Local Mona monitoring state is not
 available
                        "vmGroupName": "g1"
                    }
                ]
            }
```

```
        ]
    }
}
```

Recovering the D-MONA from One VIM Instance to Another

If a D-MONA agent fails, ESC can recover it quickly, ensuring minimal downtime and reinstating monitoring of the deployed VNFs at the earliest; However, the monitored VNFs remain unmonitored while the agent is recovered. The VNFs monitored by D-MONA remain in the last known state until the D-MONA becomes active again.

The VNFs monitoring status restores only if D-MONA successfully recovers with ESC while reprogramming each VNF monitoring rule. Until then, the D-MONA agent's state remains *UNKNOWN*, and the VNFs' state remains *UNMONITORED.*

The following example shows the state of D-MONA and VNFs in the monitoring agent API when D-MONA is down:

```
{
    "name": "Test-dmona-dep-1",
    "notificationUrl": "",
    "oneToOneMapping": false,
    "state": "UNKNOWN",
    "uri": "https://172.29.0.15:8443/mona/v1/rules",
    "vimId": "default_openstack_vim",
    "vnfData": [
        {
            "deploymentExternalId": "70d7f1f0-362e-4d2b-a89b-4877d8bfabf4",
            "deploymentName": "Test-dep-2",
            "state": "UNMONITORED",
            "vmGroupName": "g1"
        }
    ]
}
```

Disaster Recovery of a Monitoring Agent

If the deployed distributed D-MONA is unreachable due to the unavailability of the VIM, then the D-MONA can be recovered on another VIM by sending a *HealVnfRequest* with the cause `VIM_FAILURE`

Use the following to recover D-MONA on another VIM instance:

- Initiate a manual *HealVnfRequest,* as per the following SOL003 example:

Method type :

```
POST
```

VNFM Endpoint:

```
/vnf_instances/{vnfInstanceId}/heal
```

HTTP Request Header:

```
Content-Type:application/json
```

Request Payload (ETSI data structure: HealVnfRequest):

```
{
  "cause":"VIM_FAILURE"
}
```

- The recovery request is rejected if the Grant from the NFVO does not include the new *vimConnectionInfo* which identifies the VIM to redeploy the D-MONA VNF.
- When the *HealVnfRequest* completes successfully, the D-MONA VNF recreates in the new VIM and continues to monitor all the VNFs it had previously

Note The original deployment is not removed from the old VIM. Manually remove the previous D-MONA once the old VIM is reachable.

Failover in Active/Active HA

A failover in an ESC Active/Active HA deployment transfers the VNFs owned by the failed ESC instance to other ESC instances in the cluster.

If a D-MONA deployment transfers from the failed ESC instance, then it updates to UNKNOWN state in the monitoring agent API. VNFs monitored by transferred D-MONA reconciles when the D-MONA monitoring agent state is ACTIVE.

As per any D-MONA deployment, the VNFs monitored by transferred D-MONA stay in the last known state until the D-MONA becomes active once again.

Retrieving D-MONA Logs

Access the D-MONA with the `vm_credentials` password that was provided as part of the D-MONA day-0 configuration.

To retrieve the D-MONA logs, use the following command:

```
<security_user_name>:<security_user_password>
```

Where ip-address is the IP Address of the targeted D-MONA and username, password are the username and password provided as day-0 configuration at deployment of the D-MONA.

For complete list of all ESC logs, see ESC Logs section in the ESC Administration Guide.

For ETSI-related information, see Monitoring VNF Using D-MONA chapter in the Cisco Elastic Services Controller ETSI NFV MANO User Guide.

Resetting the Monitoring Rules for D-MONA

The Monitoring and Action (MONA) monitors and executes actions, such as ping, custom_scripts, and so on at specific intervals to maintain accuracy.

The local MONA keeps a track of the last known startup time of the polled D-MONA process. The status code 200 indicates the successful request. In case of successful request, the local MONA compares the last known startup time with the returned startup time from the polled application. On DMONA restart, the recovery setup starts automatically.

To enable the start time check, you must set `application_startup_time` in the dep.xml.

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 DMONA deployment.

Note Backward compatibility is not supported. It must be set for version 5.3 and later only.

Following is a sample deployment model for D-MONA:

```
<?xml version="1.0"?>
<esc_datamodel xmlns="http://www.cisco.com/esc/esc">
  <tenants>
    <tenant>
      <name>A_tenant_name</name>
      <deployments>
        <deployment>
          <name>dmona_deployment</name>
          <vm_group>
            <name>g1</name>
            <image>ESC-5_3_0_31</image>
            <flavor>m1.large</flavor>
            <bootup_time>120</bootup_time>
            <recovery_wait_time>0</recovery_wait_time>
            <interfaces>
              <interface>
                <nicid>0</nicid>
                <network>esc-net</network>
              </interface>
            </interfaces>
            <kpi_data>
              <kpi>
                <event_name>VM_ALIVE</event_name>
                <metric_value>1</metric_value>
                <metric_cond>GT</metric_cond>
                <metric_type>UINT32</metric_type>
                <metric_occurrences_true>1</metric_occurrences_true>
                <metric_occurrences_false>5</metric_occurrences_false>
                <metric_collector>
                  <type>HTTPGET</type>
                  <nicid>0</nicid>
                  <poll_frequency>3</poll_frequency>
                  <polling_unit>seconds</polling_unit>
                  <continuous_alarm>false</continuous_alarm>
                  <properties>
                 <!-- Set to true to enable start time check --->
                <property>
                <name>application_startup_time</name>
                <value>true</value>
                </property>
                 <property>
                      <name>protocol</name>
                      <value>https</value>
                   </property>
                   <property>
                     <name>port</name>
                     <value>8443</value>
                   </property>
                   <property>
                     <name>path</name>
                     <value>mona/v1/health/status</value>
                   </property>
```

```
                    </properties>
                  </metric_collector>
                </kpi>

              </kpi_data>
[...]
           </vm_group>
         </deployment>
       </deployments>
     </tenant>
   </tenants>
</esc_datamodel>
```