仮想ネットワーク機能記述子について

仮想ネットワーク機能記述子の概要

ESC は、VNF プロパティを記述する TOSCA ベースの仮想ネットワーク機能記述子（VNFD）をサポートします。VNFD は、ETSI で指定された *GS NFV-SOL 001* 仕様および標準に準拠しています。

VNFD ファイルには、VNF のインスタンス化パラメータと操作の動作が記述されています。これには、KPI、およびオンボーディングと VNF のライフサイクル管理のプロセスで使用できるその他の主要な要件が含まれています。

VNF ライフサイクル操作については、VNF ライフサイクル操作を参照してください。

仮想ネットワーク機能記述子への拡張定義

ESC はシスコが定義した VNFD の拡張機能を実装しており、ESC ではサポートされているものの ETSI 標準にはない、より高度な概念を公開しています。これらの拡張は、オーバーライドされるデータ、ノード、およびインターフェイスタイプを記述するために、シスコのタイプ定義に厳密に入力されます。

VNF の設定可能なプロパティ

VNF ノードタイプは、VNF ごとに常にスタマイズされます。シスコの拡張機能は、ESC が緩和アクションを考慮する前に、VNF が復旧するまでの復旧ポリシーと待機時間を指定する機能を提供します。

次に例を示します。

```
vnf:
      type: cisco.VPC.1_0.1_0
      properties:
        descriptor_id: b98450dd-f532-4a42-8419-e3dc04327318
        descriptor_version: '3.8'
        provider: cisco
```

```
 product_name: VPC
 software_version: 1.0
product_info_name: 'Virtual Packet Core (VPC); 32 vCPUs, 64Gb RAM, 66Gb vStorage'

 vnfm_info:
   - '9:Cisco Elastic Services Controller:v04.04.01'
 configurable_properties:
   is_autoscale_enabled: false
   is_autoheal_enabled: false
 lcm_operations_configuration:
   heal:
     recovery_action: REBOOT_THEN_REDEPLOY
     recovery_wait_time: 0
 flavour_id: default
 flavour_description: 'Default VNF Deployment Flavour'
```

コンピューティング

Cisco コンピューティングノードでは、拡張 ETSI データモデルを介して多くの ESC 機能を公開できます。これには、次の事項が含まれます。

- VIM 上の VNFC の自動生成された名前のオーバーライド。
- VIM フレーバ（VNFC に指定された ETS I 機能をオーバーライド）。
- このタイマーが期限切れになるまでアクションが実行されるのを防ぐため、ESCに予想されるブートアップ時間を指定。
- VNFC を展開後、実行/保存する Day-0 設定ブロックを提供。
- モニタリングエージェントを設定するための KPI パラメータと関連ルールの指定。
- VM グループ内の配置ルール。

次に例を示します。

```
vdu1:
  type: cisco.nodes.nfv.Vdu.Compute
  properties:
    name: Example VDU1
    description: Example VDU
    boot_order:
      - boot1-volume
    configurable_properties:
      additional_vnfc_configurable_properties:
        vim_flavor: Automation-Cirros-Flavor
        bootup_time: 1800
    name_override: my-vdu-1
    vdu_profile:
      min_number_of_instances: 1
      max_number_of_instances: 1
      static_ip_address_pool:
        network: esc-net
        ip_address_range:
          start: { get_input: VDU1_NETWORK_START }
          end: { get_input: VDU1_NETWORK_END }
        ip_addresses: { get_input: VDU1_SCALE_IP_LIST }
    kpi_data:
      VM_ALIVE-1:
       event_name: 'VM_ALIVE-1'
        metric_value: 1
        metric_cond: 'GT'
```

```
        metric_type: 'UINT32'
        metric_occurrences_true: 1
        metric_occurrences_false: 30
        metric_collector:
          type: 'ICMPPing'
          nicid: 1
          poll_frequency: 10
          polling_unit: 'seconds'
          continuous_alarm: false
    admin_rules:
      VM_ALIVE-1:
        event_name: 'VM_ALIVE-1'
        action:
          - 'ALWAYS log'
          - 'FALSE recover autohealing'
          - 'TRUE esc_vm_alive_notification'
    placement_type: zone
    placement_target: nova
    placement_enforcement: strict
    vendor_section:
      cisco_esc:
        config_data:
          example.txt:
            file: ../Files/Scripts/example.txt
            variables:
              DOMAIN_NAME: { get_input: DOMAIN_NAME }
              NAME_SERVER: { get_input: NAME_SERVER }
              VIP_ADDR: { get_input: VIP_ADDR }
              VIP_PREFIX: { get_input: VIP_PREFIX }
  capabilities:
    virtual_compute:
      properties:
        virtual_cpu:
          num_virtual_cpu: 8
        virtual_memory:
          virtual_mem_size: 16
  requirements:
    - virtual_storage:  cdr1-volume
    - virtual_storage:  boot1-volume
```

（注） 多数の入力パラメータを指定できるため、複数の展開で1つのテンプレートを使用できます。

接続ポイント

VduCp ノードタイプへのシスコの拡張機能により、主にIPアドレッシング機能とインターフェイスへのアクセシビリティが向上します。接続ポイントに追加された機能は次のとおりです。

- VIM 上のポートの自動生成された名前のオーバーライド
- 静的 IP アドレス（およびスケーリング用のプール）
- ポートが管理ポート（モニタリングに使用する）かどうかの識別
- 許可されるアドレスペア
- 特定のネットワークカードタイプとインターフェイスタイプ（SR-IOV など）のサポート
- ポート バインディング プロファイルのサポート

- ポートセキュリティが有効かどうか

次に例を示します。

```
vdu1_nic0:
      type: cisco.nodes.nfv.VduCp
      properties:
        layer_protocols: [ ipv6 ]
        protocol:
          - associated_layer_protocol: ipv6
        trunk_mode: false
        order: 0
        nw_card_model: virtio
        iface_type: direct
        management: true
        name_override: my-vdu1-nic0
        ip_subnet:
          - ip_address: { get_input: VDU1_NIC0_IP }
        allowed_address_pairs:
          - ip_address: { get_input: VDU1_NIC0_AADR_PAIRS }
        port_security_enabled: false
        binding_profile:
        trusted: true
        requirements:
        - virtual_binding: vdu1
```

展開ごとにこれらのプロパティを制御する必要がある場合は、着信要求で additionalParams として指定できる VNFD の入力で、ハードコードされた値を置き換えます。

（注） ポートバインディングプロファイルは、OpenStack の Pike 以上のバージョンで使用できます。

ボリューム

ESC は、シスコの拡張としてアウトオブバンドボリュームをサポートします。これにより、永続的なボリューム UUID の仕様を、cisco.nodes.nfv.Vdu.VirtualBlockStorage ノードに対する resourceId プロパティとして、VNFD で定義されたエフェメラルボリュームの代わりに使用できます。余分なプロパティを追加する代わりに、ESC は VIM からの UUID で識別することによって、VNFD で指定されたボリュームをオーバーライドし、独自の永続的な（展開されたアウトオブバンド）ストレージを指定することを許可します。

次に例を示します。

```
    boot1-volume:
      type: cisco.nodes.nfv.Vdu.VirtualBlockStorage
      properties:
        resource_id: { get_input: VDU1_BOOT_VOL_UUID }
        virtual_block_storage_data:
          size_of_storage: 4GB
          vdu_storage_requirements:
            vol_id: 1
            bus: ide
            type: LUKS
        sw_image_data:
          name: 'Automation_Cirros'
          version: '1.0'
          checksum: 9af30fce37a4c5c831e095745744d6d2
          container_format: bare
          disk_format: qcow2
```

```
          min_disk: 2 GB
          size: 2 GB
      artifacts:
        sw_image:
          type: tosca.artifacts.nfv.SwImage
          file: ../Files/Images/Automation-Cirros.qcow2
```

エフェメラルリソースの代わりにアウトオブバンドリソースを指定するため、ESCではインスタンス化の最中に、着信要求を使用してVNFD内のタグを照合できます。新しいデータ構造が既存の InstantiateVnfRequest に追加されます。

次の例を参考にしてください。

```
{
  "flavourId": "default",
  "instantiationLevelId": "default",
  "extVirtualLinks": [{}],
  "extManagedVirtualLinks": [{}],
  "extManagedVolumes": [
        {
            "virtualStorageDescId": "cf-cdr1-volume",
            "resourceId": "vol123"
        },
        {
            "virtualStorageDescId": "cf-boot1-volume",
            "resourceId": "vol456"
        }
    ],
  ...
}
```

（注） VNFD または InstantiateVnfRequest は、MiB、GiB、TiB 相当などのメビバイト単位のボリュームまたはソフトウェアイメージサイズを受け入れます。ボリュームまたはソフトウェアイメージのサイズが MB、GB、TB などのメガバイト単位の場合、ESC はサイズをメビバイト単位に変換し、最も近い値に調整します。わかりやすくするために、ボリュームまたはソフトウェアイメージのサイズには、必ずメビバイト単位を使用してください。

セキュリティグループルールの作成

上記のボリュームの処理に従って、ESC は VNFD で設定する代わりに、アウトオブバンド セキュリティ グループを指定する機能を提供します。これは、標準のドキュメントでセキュリティグループを説明するために使用される動詞が、非常に複雑な設定に対しては単純すぎるためです。

次に例を示します。

```
    - NETWORK_ORCH_SEC_GRP_1:
        type: cisco.policies.nfv.SecurityGroupRule
        group_name: { get_input: VIM_NETWORK_ORCH_SEC_GRP_1 }
        targets: [ vdu1_nic0 ]
```

カスタム VM 名

シスコの拡張機能では、追加パラメータを使用して、展開時の VNFC（VM）名をカスタマイズできます。ESC ETSI には、VM 名をカスタマイズするための追加パラメータが含まれています。

VIM で VM 名を設定するには、最初にデータタイプを定義してから、コンピューティングノードの Cisco ノードタイプを拡張する必要があります。

```
tosca_definitions_version: tosca_simple_yaml_1_2
data_types:
  cisco.datatypes.nfv.VnfcAdditionalConfigurableProperties:
    derived_from: tosca.datatypes.nfv.VnfcAdditionalConfigurableProperties
    properties:
      vim_flavor:
        type: string
        required: true
      bootup_time:
        type: integer
        required: true
      vm_name_override:
        type: string
        required: false
```

これらの定義により、VNFD node_templates は入力を使用してコンピューティングノードにマッピングできます。

```
topology_template:
  inputs:
    ...

  node_templates:

######################
# VDU configuration #
######################
    c1:
      type: cisco.nodes.nfv.Vdu.Compute
      properties:
        name: control-function 1
        description: Vdu1 of an active:standby (1:1) redundant pair of CF VMs
        ...
        configurable_properties:
          additional_vnfc_configurable_properties:
            vim_flavor: { get_input: CF_FLAVOR }
            bootup_time: { get_input: BOOTUP_TIME_CF }
            vm_name_override: { get_input: VIM_C1_VM_NAME ] }
        ...
      capabilities:
        virtual_compute:
          properties:
            virtual_cpu:
              num_virtual_cpu: 8
            virtual_memory:
              virtual_mem_size: 16 GiB
      requirements:
        - virtual_storage:  cf-cdr1-volume
        - virtual_storage:  cf-boot1-volume
```

コンピューティングノードの設定可能なプロパティで、*vm_name_override* を指定します。*vm_name_override* が指定されていない場合、ESC によって VM 名が自動生成されます。

ESCは、VNFC を表すコンピューティングノードに与えられたラベルと一致する vduId によって識別される VNFC の *VnfInstance.instantiatedVnfInfo.vnfcResourceInfo.metadata.vim_vm_name* に VNFC 固有の値を保存します。

ライフサイクル管理操作の詳細については、VNF ライフサイクルの管理を参照してください。

SR-IOV

ESC ETSI NFV MANO は、Cisco データタイプを使用する SR-IOV プロパティをサポートします。VNFC を SR-IOV パススルーアダプタに関連付けるよう、インターフェイスを設定できます。

Cisco データタイプ：

```
cisco.datatypes.nfv.L2ProtocolData:
  derived_from: tosca.datatypes.nfv.L2ProtocolData
  properties:
    segmentation_id:
      type: integer
      required: false
```

VNFD の例：

```
  virtual_link_protocol_data:
  - associated_layer_protocol: ethernet
    l2_protocol_data:
      network_type: vlan
      physical_network: vlan_network
      segmentation_id: { get_input: VL1_SEG_ID }
```