



## **Cisco Elastic Service Controller 2.0 REST API Guide**

**Cisco Systems, Inc.**

[www.cisco.com](http://www.cisco.com)

Cisco has more than 200 offices worldwide.  
Addresses, phone numbers, and fax numbers  
are listed on the Cisco website at

[www.cisco.com/go/offices](http://www.cisco.com/go/offices).

Cisco Elastic Service Controller 2.0 REST API Guide  
© 1999–2015 Cisco Systems, Inc. All rights reserved.

## Deployment Operations

POST

/v0/{internal\_tenant\_id}/deployments/service/{internal\_deployment\_id}

Handle deployment operation request with the operation type as payload

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internal_tenant_id</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID</b>	path	string
<b>internal_deployment_id</b>	<input type="text" value="(required)"/>	<b>Internal deployment ID</b>	path	string
<b>body</b>	<input type="text" value="(required)"/> Parameter content type: <input type="text" value="application/xml"/>	<b>Deployment operation payload specifying the type of operation</b>	body	<p>Model   Model Schema</p> <pre>{   "operation": "string" }</pre> <p>Click to set as parameter value</p>

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Deployment operation request accepted		
400	Missing callback		
404	Deployment or tenant not found		

Try it out!

## Deployments

GET

/v0/deployments

Find all deployments

## Implementation Notes

All deployments created in ESC will be returned as a List

## Response Class (Status 200)

Model | Model Schema

```
[
  {
    "dbDatamodel": {
      "services": {
        "service_definition": [
          {
            "deployment_name": "string",
            "name": "string",
            "version": "string",
            "policies": {
              "external_dependencies": {
                "service_dependency": {
                  "service": [
                    {
                      "service_name": "string",
                      "service_version": "string"
                    }
                  ]
                },
              "network_dependency": {
                "network": [
                  "string"
                ]
              }
            },
            "placement": [
              {
                "type": "AFFINITY",
                "enforcement": "STRICT",
                "host": "string",
                "zone": "string"
              }
            ],
            "startup": [
              {
                "target_vm_group_ref": "string",
                "depends_on": [
                  {
                    "vm_group_ref": "string",
                    "kpi_event_ref": "string"
                  }
                ]
              }
            ]
          }
        ]
      }
    }
  }
]
```

```

    }
  ]
}
],
"networks": {
  "network": [
    {
      "name": "string",
      "shared": false,
      "admin_state": false,
      "router_external": false,
      "provider_physical_network": "string",
      "provider_network_type": "string",
      "provider_segmentation_id": 0,
      "subnet": [
        {
          "name": "string",
          "ipversion": "IPV_4",
          "dhcp": false,
          "address": "string",
          "netmask": "string",
          "gateway": "string"
        }
      ],
      "switch_name": "string",
      "vlan_id": 0,
      "number_of_ports": 0
    }
  ]
},
"volumes": {
  "volume": [
    {
      "external_volume_id": "string",
      "internal_volume_id": "string",
      "size": "string",
      "sizeunit": "string",
      "bus": "string"
    }
  ]
},
"vm_group": [
  {
    "name": "string",
    "image": "string",
    "flavor": "string",
    "bootup_time": 0,

```

```
"recovery_wait_time": 0,
"misc": {
  "watchdog": {
    "hardware": "I_6300_ESB",
    "action": "RESET"
  }
},
"volumes": {
  "volume": [
    {
      "external_volume_id": "string",
      "internal_volume_id": "string",
      "size": "string",
      "sizeunit": "string",
      "bus": "string"
    }
  ]
},
"interfaces": {
  "interface": [
    {
      "network_uuid": "string",
      "subnet_uuid": "string",
      "ip_address": "string",
      "mac_address": "string",
      "port_uuid": "string",
      "port_forward": [
        {
          "port_forwarding_type": "string",
          "port_number": "string"
        }
      ],
      "type": "string",
      "nic_id": 0,
      "anycast_ip": "string",
      "shared_ip": "string",
      "netmask": "string",
      "gateway": "string"
    }
  ]
},
"kpi_data": {
  "enabled": false,
  "kpi": [
    {
      "event_name": "string",
      "metric_value": "string",
      "metric_cond": "GT",
```

```
    "metric_type": "INT_8",
    "metric_occurrences_true": 0,
    "metric_occurrences_false": 0,
    "metric_proxy": "string",
    "metric_collector": {
      "type": "string",
      "nicid": 0,
      "proxy": "string",
      "poll_frequency": 0,
      "polling_unit": "MINUTES",
      "continuous_alarm": false
    }
  }
],
},
"rules": {
  "admin_rules": {
    "rule": [
      {
        "event_name": "string",
        "action": [
          "string"
        ]
      }
    ]
  },
  "user_rules": {
    "rule": [
      {
        "event_name": "string",
        "action": [
          "string"
        ]
      }
    ]
  }
},
"config_data": {
  "configuration": [
    {
      "dst": "string",
      "file": "string",
      "data": "string",
      "variable": [
        {
          "name": "string",
          "val": [
            "string"
          ]
        }
      ]
    }
  ]
}
```

```

        ]
    }
]
},
"port_forwarding": {
    "port": [
        {
            "type": "string",
            "protocol": "string",
            "vnf_port": 0,
            "external_port_range": [
                {
                    "start": 0,
                    "end": 0
                }
            ]
        }
    ]
},
"scaling": {
    "min_active": 0,
    "min_standby": 0,
    "max_active": 0,
    "max_standby": 0,
    "standby_ratio": "string",
    "elastic": false,
    "static_mac_address_pool": [
        {
            "network": "string",
            "mac_address_range": [
                {
                    "start": "string",
                    "end": "string"
                }
            ],
            "mac_address": [
                "string"
            ]
        }
    ],
    "static_license_pool": {
        "license": [
            {
                "license_key_value": "string",
                "license_key_type": "MAC_ADDR",
                "license_file": "string",

```



```
        "license_data": "string"
    }
]
},
"static_ip_address_pool": [
    {
        "network": "string",
        "gateway": "string",
        "netmask": "string",
        "ip_address_range": [
            {
                "start": "string",
                "end": "string"
            }
        ],
        "ip_address": [
            "string"
        ]
    }
],
"placement": [
    {
        "type": "AFFINITY",
        "enforcement": "STRICT",
        "host": "string",
        "zone": "string"
    }
],
"shared_ip": [
    {
        "nic_id": 0,
        "shared_ip": "string"
    }
],
"properties": {
    "property": [
        {
            "name": "string",
            "value": "string"
        }
    ]
},
"recovery_policy": {
    "action_on_recovery": "REBOOT_THEN_REDEPLOY"
},
"extensions": {
    "extension": [
```

```

    {
      "name": "string",
      "properties": {
        "property": [
          {
            "name": "string",
            "value": "string"
          }
        ]
      },
      "containers": {
        "container": [
          {
            "name": "string",
            "properties": {
              "property": [
                {
                  "name": "string",
                  "value": "string"
                }
              ]
            }
          }
        ]
      }
    }
  ],
},
"tenants": {
  "tenant": [
    {
      "name": "string",
      "networks": {
        "network": [
          {
            "name": "string",
            "shared": false,
            "admin_state": false,
            "router_external": false,
            "provider_physical_network": "string",
            "provider_network_type": "string",
            "provider_segmentation_id": 0,
            "subnet": [

```

```
        {
            "name": "string",
            "ipversion": "IPV_4",
            "dhcp": false,
            "address": "string",
            "netmask": "string",
            "gateway": "string"
        }
    ],
    "switch_name": "string",
    "vlan_id": 0,
    "number_of_ports": 0
}
],
},
"services": {
    "service_definition": [
        {
            "deployment_name": "string",
            "name": "string",
            "version": "string",
            "policies": {
                "external_dependencies": {
                    "service_dependency": {
                        "service": [
                            {
                                "service_name": "string",
                                "service_version": "string"
                            }
                        ]
                    },
                    "network_dependency": {
                        "network": [
                            "string"
                        ]
                    }
                }
            },
            "placement": [
                {
                    "type": "AFFINITY",
                    "enforcement": "STRICT",
                    "host": "string",
                    "zone": "string"
                }
            ],
            "startup": [
                {
                    "target_vm_group_ref": "string",
```

```
        "depends_on": [
            {
                "vm_group_ref": "string",
                "kpi_event_ref": "string"
            }
        ]
    },
    "networks": {
        "network": [
            {
                "name": "string",
                "shared": false,
                "admin_state": false,
                "router_external": false,
                "provider_physical_network": "string",
                "provider_network_type": "string",
                "provider_segmentation_id": 0,
                "subnet": [
                    {
                        "name": "string",
                        "ipversion": "IPV_4",
                        "dhcp": false,
                        "address": "string",
                        "netmask": "string",
                        "gateway": "string"
                    }
                ],
                "switch_name": "string",
                "vlan_id": 0,
                "number_of_ports": 0
            }
        ]
    },
    "volumes": {
        "volume": [
            {
                "external_volume_id": "string",
                "internal_volume_id": "string",
                "size": "string",
                "sizeunit": "string",
                "bus": "string"
            }
        ]
    },
    "vm_group": [
        {
```

```
"name": "string",
"image": "string",
"flavor": "string",
"bootup_time": 0,
"recovery_wait_time": 0,
"misc": {
  "watchdog": {
    "hardware": "I_6300_ESB",
    "action": "RESET"
  }
},
"volumes": {
  "volume": [
    {
      "external_volume_id": "string",
      "internal_volume_id": "string",
      "size": "string",
      "sizeunit": "string",
      "bus": "string"
    }
  ]
},
"interfaces": {
  "interface": [
    {
      "network_uuid": "string",
      "subnet_uuid": "string",
      "ip_address": "string",
      "mac_address": "string",
      "port_uuid": "string",
      "port_forward": [
        {
          "port_forwarding_type": "string",
          "port_number": "string"
        }
      ],
      "type": "string",
      "nic_id": 0,
      "anycast_ip": "string",
      "shared_ip": "string",
      "netmask": "string",
      "gateway": "string"
    }
  ]
},
"kpi_data": {
  "enabled": false,
  "kpi": [
```

```
{
  "event_name": "string",
  "metric_value": "string",
  "metric_cond": "GT",
  "metric_type": "INT_8",
  "metric_occurrences_true": 0,
  "metric_occurrences_false": 0,
  "metric_proxy": "string",
  "metric_collector": {
    "type": "string",
    "nicid": 0,
    "proxy": "string",
    "poll_frequency": 0,
    "polling_unit": "MINUTES",
    "continuous_alarm": false
  }
}
],
"rules": {
  "admin_rules": {
    "rule": [
      {
        "event_name": "string",
        "action": [
          "string"
        ]
      }
    ]
  },
  "user_rules": {
    "rule": [
      {
        "event_name": "string",
        "action": [
          "string"
        ]
      }
    ]
  }
},
"config_data": {
  "configuration": [
    {
      "dst": "string",
      "file": "string",
      "data": "string",
      "variable": [
```

```
        {
            "name": "string",
            "val": [
                "string"
            ]
        }
    ]
},
"port_forwarding": {
    "port": [
        {
            "type": "string",
            "protocol": "string",
            "vnf_port": 0,
            "external_port_range": [
                {
                    "start": 0,
                    "end": 0
                }
            ]
        }
    ]
},
"scaling": {
    "min_active": 0,
    "min_standby": 0,
    "max_active": 0,
    "max_standby": 0,
    "standby_ratio": "string",
    "elastic": false,
    "static_mac_address_pool": [
        {
            "network": "string",
            "mac_address_range": [
                {
                    "start": "string",
                    "end": "string"
                }
            ],
            "mac_address": [
                "string"
            ]
        }
    ],
    "static_license_pool": {
        "license": [
```

```
        {
            "license_key_value": "string",
            "license_key_type": "MAC_ADDR",
            "license_file": "string",
            "license_data": "string"
        }
    ]
},
"static_ip_address_pool": [
    {
        "network": "string",
        "gateway": "string",
        "netmask": "string",
        "ip_address_range": [
            {
                "start": "string",
                "end": "string"
            }
        ],
        "ip_address": [
            "string"
        ]
    }
]
},
"placement": [
    {
        "type": "AFFINITY",
        "enforcement": "STRICT",
        "host": "string",
        "zone": "string"
    }
],
"shared_ip": [
    {
        "nic_id": 0,
        "shared_ip": "string"
    }
],
"properties": {
    "property": [
        {
            "name": "string",
            "value": "string"
        }
    ]
},
"recovery_policy": {
```



```

        "action_on_recovery": "REBOOT_THEN_REDEPLOY"
    },
    "extensions": {
        "extension": [
            {
                "name": "string",
                "properties": {
                    "property": [
                        {
                            "name": "string",
                            "value": "string"
                        }
                    ]
                }
            },
            {
                "containers": {
                    "container": [
                        {
                            "name": "string",
                            "properties": {
                                "property": [
                                    {
                                        "name": "string",
                                        "value": "string"
                                    }
                                ]
                            }
                        }
                    ]
                }
            }
        ]
    },
    "deployments": {
        "deployment": [
            {
                "name": "string",
                "policies": {
                    "external_dependencies": {
                        "service_dependency": {
                            "service": [
                                {
                                    "service_name": "string",
                                    "service_version": "string"
                                }
                            ]
                        }
                    }
                }
            }
        ]
    }
}

```

```

    }
  ]
},
"network_dependency": {
  "network": [
    "string"
  ]
}
},
"placement": [
  {
    "type": "AFFINITY",
    "enforcement": "STRICT",
    "host": "string",
    "zone": "string"
  }
],
"startup": [
  {
    "target_vm_group_ref": "string",
    "depends_on": [
      {
        "vm_group_ref": "string",
        "kpi_event_ref": "string"
      }
    ]
  }
]
},
"networks": {
  "network": [
    {
      "name": "string",
      "shared": false,
      "admin_state": false,
      "router_external": false,
      "provider_physical_network": "string",
      "provider_network_type": "string",
      "provider_segmentation_id": 0,
      "subnet": [
        {
          "name": "string",
          "ipversion": "IPV_4",
          "dhcp": false,
          "address": "string",
          "netmask": "string",
          "gateway": "string"
        }
      ]
    }
  ]
}

```

```
    ],
    "switch_name": "string",
    "vlan_id": 0,
    "number_of_ports": 0
  }
]
},
"volumes": {
  "volume": [
    {
      "external_volume_id": "string",
      "internal_volume_id": "string",
      "size": "string",
      "sizeunit": "string",
      "bus": "string"
    }
  ]
},
"vm_group": [
  {
    "name": "string",
    "image": "string",
    "flavor": "string",
    "bootup_time": 0,
    "recovery_wait_time": 0,
    "misc": {
      "watchdog": {
        "hardware": "I_6300_ESB",
        "action": "RESET"
      }
    },
    "volumes": {
      "volume": [
        {
          "external_volume_id": "string",
          "internal_volume_id": "string",
          "size": "string",
          "sizeunit": "string",
          "bus": "string"
        }
      ]
    },
    "interfaces": {
      "interface": [
        {
          "network_uuid": "string",
          "subnet_uuid": "string",
          "ip_address": "string",
```

```
    "mac_address": "string",
    "port_uuid": "string",
    "port_forward": [
      {
        "port_forwarding_type": "string",
        "port_number": "string"
      }
    ],
    "type": "string",
    "nic_id": 0,
    "anycast_ip": "string",
    "shared_ip": "string",
    "netmask": "string",
    "gateway": "string"
  }
]
},
"kpi_data": {
  "enabled": false,
  "kpi": [
    {
      "event_name": "string",
      "metric_value": "string",
      "metric_cond": "GT",
      "metric_type": "INT_8",
      "metric_occurrences_true": 0,
      "metric_occurrences_false": 0,
      "metric_proxy": "string",
      "metric_collector": {
        "type": "string",
        "nicid": 0,
        "proxy": "string",
        "poll_frequency": 0,
        "polling_unit": "MINUTES",
        "continuous_alarm": false
      }
    }
  ]
},
"rules": {
  "admin_rules": {
    "rule": [
      {
        "event_name": "string",
        "action": [
          "string"
        ]
      }
    ]
  }
}
```

```
]
},
"user_rules": {
  "rule": [
    {
      "event_name": "string",
      "action": [
        "string"
      ]
    }
  ]
},
"config_data": {
  "configuration": [
    {
      "dst": "string",
      "file": "string",
      "data": "string",
      "variable": [
        {
          "name": "string",
          "val": [
            "string"
          ]
        }
      ]
    }
  ]
},
"port_forwarding": {
  "port": [
    {
      "type": "string",
      "protocol": "string",
      "vnf_port": 0,
      "external_port_range": [
        {
          "start": 0,
          "end": 0
        }
      ]
    }
  ]
},
"scaling": {
  "min_active": 0,
  "min_standby": 0,
```

```
"max_active": 0,
"max_standby": 0,
"standby_ratio": "string",
"elastic": false,
"static_mac_address_pool": [
  {
    "network": "string",
    "mac_address_range": [
      {
        "start": "string",
        "end": "string"
      }
    ],
    "mac_address": [
      "string"
    ]
  }
],
"static_license_pool": {
  "license": [
    {
      "license_key_value": "string",
      "license_key_type": "MAC_ADDR",
      "license_file": "string",
      "license_data": "string"
    }
  ]
},
"static_ip_address_pool": [
  {
    "network": "string",
    "gateway": "string",
    "netmask": "string",
    "ip_address_range": [
      {
        "start": "string",
        "end": "string"
      }
    ],
    "ip_address": [
      "string"
    ]
  }
],
"placement": [
  {
    "type": "AFFINITY",
```

```
        "enforcement": "STRICT",
        "host": "string",
        "zone": "string"
    }
],
"shared_ip": [
    {
        "nic_id": 0,
        "shared_ip": "string"
    }
],
"properties": {
    "property": [
        {
            "name": "string",
            "value": "string"
        }
    ]
},
"recovery_policy": {
    "action_on_recovery": "REBOOT_THEN_REDEPLOY"
},
"extensions": {
    "extension": [
        {
            "name": "string",
            "properties": {
                "property": [
                    {
                        "name": "string",
                        "value": "string"
                    }
                ]
            }
        }
    ],
    "containers": {
        "container": [
            {
                "name": "string",
                "properties": {
                    "property": [
                        {
                            "name": "string",
                            "value": "string"
                        }
                    ]
                }
            }
        ]
    }
}
```

```

    }
  }
]
},
"networks": {
  "network": [
    {
      "name": "string",
      "shared": false,
      "admin_state": false,
      "router_external": false,
      "provider_physical_network": "string",
      "provider_network_type": "string",
      "provider_segmentation_id": 0,
      "subnet": [
        {
          "name": "string",
          "ipversion": "IPV_4",
          "dhcp": false,
          "address": "string",
          "netmask": "string",
          "gateway": "string"
        }
      ],
      "switch_name": "string",
      "vlan_id": 0,
      "number_of_ports": 0
    }
  ]
},
"images": {
  "image": [
    {
      "name": "string",
      "src": "string",
      "size": 0,
      "sizeunit": "MI_B",
      "disk_format": "QCOW_2",
      "container_format": "string",
      "serial_console": false,

```




```
        "e1000_net": false,
        "virtio_net": false,
        "disk_bus": "IDE"
    }
]
},
"flavors": {
    "flavor": [
        {
            "name": "string",
            "vcpus": 0,
            "sockets": 0,
            "cores": 0,
            "threads": 0,
            "memory_mb": 0,
            "root_disk_mb": 0,
            "ephemeral_disk_mb": 0,
            "swap_disk_mb": 0
        }
    ]
}
},
"dbDeployment": {
    "deploymentId": "string",
    "deploymentName": "string",
    "tenantId": "string",
    "serviceRegistrationId": "string",
    "eventTime": "string",
    "eventNotificationContentType": 0,
    "eventNotificationType": 0,
    "eventNotificationUrl": "string",
    "dataModel": "string",
    "classMetadata": {
        "entityName": "string",
        "propertyNames": [
            "string"
        ],
        "naturalIdentifierProperties": [
            0
        ],
        "identifierType": {
            "associationType": false,
            "entityType": false,
            "anyType": false,
            "componentType": false,
            "xmlelement": false,
            "mutable": false,
            "collectionType": false,
```

```
    "name": "string"
  },
  "identifierPropertyName": "string",
  "propertyTypes": [
    {
      "associationType": false,
      "entityType": false,
      "anyType": false,
      "componentType": false,
      "xmlElement": false,
      "mutable": false,
      "collectionType": false,
      "name": "string"
    }
  ],
  "mutable": false,
  "propertyNullability": [
    true
  ],
  "versioned": false,
  "versionProperty": 0,
  "propertyLaziness": [
    true
  ],
  "inherited": false
},
"id": "string"
},
"deployment_details": [
  {
    "sharedIps": [
      {
        "nic_id": 0,
        "shared_ip": "string"
      }
    ],
    "vm_uuid": "string",
    "host_uuid": "string",
    "host_name": "string",
    "vm_name": "string",
    "vm_group_name": "string",
    "volume": [
      {
        "external_volume_id": "string",
        "internal_volume_id": "string",
        "size": "string",
        "sizeunit": "string",
        "bus": "string"
      }
    ]
  }
]
```

```

    }
  ],
  "interface": [
    {
      "network_uuid": "string",
      "subnet_uuid": "string",
      "ip_address": "string",
      "mac_address": "string",
      "port_uuid": "string",
      "port_forward": [
        {
          "port_forwarding_type": "string",
          "port_number": "string"
        }
      ],
      "type": "string",
      "nic_id": 0,
      "anycast_ip": "string",
      "shared_ip": "string",
      "netmask": "string",
      "gateway": "string"
    }
  ],
  "vm_state_machine_state": "string"
}
],
"internal_tenant_id": "string",
"external_tenant_id": "string",
"internal_service_registration_id": "string",
"internal_deployment_id": "string",
"external_service_registration_id": "string",
"external_deployment_id": "string",
"deployment_stage": "string",
"deployment_name": "string",
"service_name": "string",
"service_version": "string",
"service_state_machine_state": "string"
}
]

```

Response Content Type  

Try it out!

**DELETE** /v0/deployments/{internal\_deployment\_id}

Delete a deployment from ESC

## Implementation Notes

Delete a deployment from ESC

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internal_deployment_id</b>	<input type="text" value="(required)"/>	<b>Internal ID of the deployment</b>	path	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
default	successful operation		

[Try it out!](#)

**GET** /v0/deployments/{internal\_deployment\_id}

[Find a deployment by its internal ID](#)

## Implementation Notes

A deployment will be returned if found in ESC

## Response Class (Status 200)

Model | **Model Schema**

```
{
  "dbDatamodel": {
    "services": {
      "service_definition": [
        {
          "deployment_name": "string",
          "name": "string",
          "version": "string",
          "policies": {
            "external_dependencies": {
              "service_dependency": {
                "service": [
                  {
                    "service_name": "string",
                    "service_version": "string"
                  }
                ]
              }
            }
          }
        }
      ]
    }
  }
}
```

```

    ]
  },
  "network_dependency": {
    "network": [
      "string"
    ]
  }
},
"placement": [
  {
    "type": "AFFINITY",
    "enforcement": "STRICT",
    "host": "string",
    "zone": "string"
  }
],
"startup": [
  {
    "target_vm_group_ref": "string",
    "depends_on": [
      {
        "vm_group_ref": "string",
        "kpi_event_ref": "string"
      }
    ]
  }
]
},
"networks": {
  "network": [
    {
      "name": "string",
      "shared": false,
      "admin_state": false,
      "router_external": false,
      "provider_physical_network": "string",
      "provider_network_type": "string",
      "provider_segmentation_id": 0,
      "subnet": [
        {
          "name": "string",
          "ipversion": "IPV_4",
          "dhcp": false,
          "address": "string",
          "netmask": "string",
          "gateway": "string"
        }
      ]
    }
  ],

```

```
    "switch_name": "string",
    "vlan_id": 0,
    "number_of_ports": 0
  }
]
},
"volumes": {
  "volume": [
    {
      "external_volume_id": "string",
      "internal_volume_id": "string",
      "size": "string",
      "sizeunit": "string",
      "bus": "string"
    }
  ]
},
"vm_group": [
  {
    "name": "string",
    "image": "string",
    "flavor": "string",
    "bootup_time": 0,
    "recovery_wait_time": 0,
    "misc": {
      "watchdog": {
        "hardware": "I_6300_ESB",
        "action": "RESET"
      }
    },
    "volumes": {
      "volume": [
        {
          "external_volume_id": "string",
          "internal_volume_id": "string",
          "size": "string",
          "sizeunit": "string",
          "bus": "string"
        }
      ]
    },
    "interfaces": {
      "interface": [
        {
          "network_uuid": "string",
          "subnet_uuid": "string",
          "ip_address": "string",
          "mac_address": "string",
```

```
    "port_uuid": "string",
    "port_forward": [
      {
        "port_forwarding_type": "string",
        "port_number": "string"
      }
    ],
    "type": "string",
    "nic_id": 0,
    "anycast_ip": "string",
    "shared_ip": "string",
    "netmask": "string",
    "gateway": "string"
  }
]
},
"kpi_data": {
  "enabled": false,
  "kpi": [
    {
      "event_name": "string",
      "metric_value": "string",
      "metric_cond": "GT",
      "metric_type": "INT_8",
      "metric_occurrences_true": 0,
      "metric_occurrences_false": 0,
      "metric_proxy": "string",
      "metric_collector": {
        "type": "string",
        "nicid": 0,
        "proxy": "string",
        "poll_frequency": 0,
        "polling_unit": "MINUTES",
        "continuous_alarm": false
      }
    }
  ]
}
},
"rules": {
  "admin_rules": {
    "rule": [
      {
        "event_name": "string",
        "action": [
          "string"
        ]
      }
    ]
  }
}
```

```
    },
    "user_rules": {
      "rule": [
        {
          "event_name": "string",
          "action": [
            "string"
          ]
        }
      ]
    }
  },
  "config_data": {
    "configuration": [
      {
        "dst": "string",
        "file": "string",
        "data": "string",
        "variable": [
          {
            "name": "string",
            "val": [
              "string"
            ]
          }
        ]
      }
    ]
  },
  "port_forwarding": {
    "port": [
      {
        "type": "string",
        "protocol": "string",
        "vnf_port": 0,
        "external_port_range": [
          {
            "start": 0,
            "end": 0
          }
        ]
      }
    ]
  },
  "scaling": {
    "min_active": 0,
    "min_standby": 0,
    "max_active": 0,
```



```
"max_standby": 0,
"standby_ratio": "string",
"elastic": false,
"static_mac_address_pool": [
  {
    "network": "string",
    "mac_address_range": [
      {
        "start": "string",
        "end": "string"
      }
    ],
    "mac_address": [
      "string"
    ]
  }
],
"static_license_pool": {
  "license": [
    {
      "license_key_value": "string",
      "license_key_type": "MAC_ADDR",
      "license_file": "string",
      "license_data": "string"
    }
  ]
},
"static_ip_address_pool": [
  {
    "network": "string",
    "gateway": "string",
    "netmask": "string",
    "ip_address_range": [
      {
        "start": "string",
        "end": "string"
      }
    ],
    "ip_address": [
      "string"
    ]
  }
],
"placement": [
  {
    "type": "AFFINITY",
    "enforcement": "STRICT",
```

```
        "host": "string",
        "zone": "string"
    }
],
"shared_ip": [
    {
        "nic_id": 0,
        "shared_ip": "string"
    }
],
"properties": {
    "property": [
        {
            "name": "string",
            "value": "string"
        }
    ]
},
"recovery_policy": {
    "action_on_recovery": "REBOOT_THEN_REDEPLOY"
},
"extensions": {
    "extension": [
        {
            "name": "string",
            "properties": {
                "property": [
                    {
                        "name": "string",
                        "value": "string"
                    }
                ]
            }
        }
    ],
    "containers": {
        "container": [
            {
                "name": "string",
                "properties": {
                    "property": [
                        {
                            "name": "string",
                            "value": "string"
                        }
                    ]
                }
            }
        ]
    }
}
```

```

    }
  ]
}
]
},
"tenants": {
  "tenant": [
    {
      "name": "string",
      "networks": {
        "network": [
          {
            "name": "string",
            "shared": false,
            "admin_state": false,
            "router_external": false,
            "provider_physical_network": "string",
            "provider_network_type": "string",
            "provider_segmentation_id": 0,
            "subnet": [
              {
                "name": "string",
                "ipversion": "IPV_4",
                "dhcp": false,
                "address": "string",
                "netmask": "string",
                "gateway": "string"
              }
            ],
            "switch_name": "string",
            "vlan_id": 0,
            "number_of_ports": 0
          }
        ]
      },
    },
    "services": {
      "service_definition": [
        {
          "deployment_name": "string",
          "name": "string",
          "version": "string",
          "policies": {
            "external_dependencies": {
              "service_dependency": {
                "service": [

```

```
        {
            "service_name": "string",
            "service_version": "string"
        }
    ]
},
"network_dependency": {
    "network": [
        "string"
    ]
}
},
"placement": [
    {
        "type": "AFFINITY",
        "enforcement": "STRICT",
        "host": "string",
        "zone": "string"
    }
],
"startup": [
    {
        "target_vm_group_ref": "string",
        "depends_on": [
            {
                "vm_group_ref": "string",
                "kpi_event_ref": "string"
            }
        ]
    }
]
},
"networks": {
    "network": [
        {
            "name": "string",
            "shared": false,
            "admin_state": false,
            "router_external": false,
            "provider_physical_network": "string",
            "provider_network_type": "string",
            "provider_segmentation_id": 0,
            "subnet": [
                {
                    "name": "string",
                    "ipversion": "IPV_4",
                    "dhcp": false,
                    "address": "string",
```

```
        "netmask": "string",
        "gateway": "string"
    }
],
"switch_name": "string",
"vlan_id": 0,
"number_of_ports": 0
}
]
},
"volumes": {
    "volume": [
        {
            "external_volume_id": "string",
            "internal_volume_id": "string",
            "size": "string",
            "sizeunit": "string",
            "bus": "string"
        }
    ]
},
"vm_group": [
    {
        "name": "string",
        "image": "string",
        "flavor": "string",
        "bootup_time": 0,
        "recovery_wait_time": 0,
        "misc": {
            "watchdog": {
                "hardware": "I_6300_ESB",
                "action": "RESET"
            }
        },
        "volumes": {
            "volume": [
                {
                    "external_volume_id": "string",
                    "internal_volume_id": "string",
                    "size": "string",
                    "sizeunit": "string",
                    "bus": "string"
                }
            ]
        },
        "interfaces": {
            "interface": [
                {
```

```
"network_uuid": "string",
"subnet_uuid": "string",
"ip_address": "string",
"mac_address": "string",
"port_uuid": "string",
"port_forward": [
  {
    "port_forwarding_type": "string",
    "port_number": "string"
  }
],
"type": "string",
"nic_id": 0,
"anycast_ip": "string",
"shared_ip": "string",
"netmask": "string",
"gateway": "string"
}
]
},
"kpi_data": {
  "enabled": false,
  "kpi": [
    {
      "event_name": "string",
      "metric_value": "string",
      "metric_cond": "GT",
      "metric_type": "INT_8",
      "metric_occurrences_true": 0,
      "metric_occurrences_false": 0,
      "metric_proxy": "string",
      "metric_collector": {
        "type": "string",
        "nicid": 0,
        "proxy": "string",
        "poll_frequency": 0,
        "polling_unit": "MINUTES",
        "continuous_alarm": false
      }
    }
  ]
}
},
"rules": {
  "admin_rules": {
    "rule": [
      {
        "event_name": "string",
        "action": [
```

```
        "string"
      ]
    }
  ],
  "user_rules": {
    "rule": [
      {
        "event_name": "string",
        "action": [
          "string"
        ]
      }
    ]
  },
  "config_data": {
    "configuration": [
      {
        "dst": "string",
        "file": "string",
        "data": "string",
        "variable": [
          {
            "name": "string",
            "val": [
              "string"
            ]
          }
        ]
      }
    ]
  },
  "port_forwarding": {
    "port": [
      {
        "type": "string",
        "protocol": "string",
        "vnf_port": 0,
        "external_port_range": [
          {
            "start": 0,
            "end": 0
          }
        ]
      }
    ]
  },
}
```

```
"scaling": {
  "min_active": 0,
  "min_standby": 0,
  "max_active": 0,
  "max_standby": 0,
  "standby_ratio": "string",
  "elastic": false,
  "static_mac_address_pool": [
    {
      "network": "string",
      "mac_address_range": [
        {
          "start": "string",
          "end": "string"
        }
      ],
      "mac_address": [
        "string"
      ]
    }
  ],
  "static_license_pool": {
    "license": [
      {
        "license_key_value": "string",
        "license_key_type": "MAC_ADDR",
        "license_file": "string",
        "license_data": "string"
      }
    ]
  },
  "static_ip_address_pool": [
    {
      "network": "string",
      "gateway": "string",
      "netmask": "string",
      "ip_address_range": [
        {
          "start": "string",
          "end": "string"
        }
      ],
      "ip_address": [
        "string"
      ]
    }
  ]
},
```



```
"placement": [
  {
    "type": "AFFINITY",
    "enforcement": "STRICT",
    "host": "string",
    "zone": "string"
  }
],
"shared_ip": [
  {
    "nic_id": 0,
    "shared_ip": "string"
  }
],
"properties": {
  "property": [
    {
      "name": "string",
      "value": "string"
    }
  ]
},
"recovery_policy": {
  "action_on_recovery": "REBOOT_THEN_REDEPLOY"
},
"extensions": {
  "extension": [
    {
      "name": "string",
      "properties": {
        "property": [
          {
            "name": "string",
            "value": "string"
          }
        ]
      }
    }
  ],
  "containers": {
    "container": [
      {
        "name": "string",
        "properties": {
          "property": [
            {
              "name": "string",
              "value": "string"
            }
          ]
        }
      }
    ]
  }
}
```

```

    }
  }
]
},
"deployments": {
  "deployment": [
    {
      "name": "string",
      "policies": {
        "external_dependencies": {
          "service_dependency": {
            "service": [
              {
                "service_name": "string",
                "service_version": "string"
              }
            ]
          },
          "network_dependency": {
            "network": [
              "string"
            ]
          }
        },
        "placement": [
          {
            "type": "AFFINITY",
            "enforcement": "STRICT",
            "host": "string",
            "zone": "string"
          }
        ],
        "startup": [
          {
            "target_vm_group_ref": "string",
            "depends_on": [
              {
                "vm_group_ref": "string",
                "kpi_event_ref": "string"
              }
            ]
          }
        ]
      }
    }
  ]
}

```

```

    ]
  }
]
},
"networks": {
  "network": [
    {
      "name": "string",
      "shared": false,
      "admin_state": false,
      "router_external": false,
      "provider_physical_network": "string",
      "provider_network_type": "string",
      "provider_segmentation_id": 0,
      "subnet": [
        {
          "name": "string",
          "ipversion": "IPV_4",
          "dhcp": false,
          "address": "string",
          "netmask": "string",
          "gateway": "string"
        }
      ],
      "switch_name": "string",
      "vlan_id": 0,
      "number_of_ports": 0
    }
  ]
},
"volumes": {
  "volume": [
    {
      "external_volume_id": "string",
      "internal_volume_id": "string",
      "size": "string",
      "sizeunit": "string",
      "bus": "string"
    }
  ]
},
"vm_group": [
  {
    "name": "string",
    "image": "string",
    "flavor": "string",
    "bootup_time": 0,
    "recovery_wait_time": 0,

```

```
"misc": {
  "watchdog": {
    "hardware": "I_6300_ESB",
    "action": "RESET"
  }
},
"volumes": {
  "volume": [
    {
      "external_volume_id": "string",
      "internal_volume_id": "string",
      "size": "string",
      "sizeunit": "string",
      "bus": "string"
    }
  ]
},
"interfaces": {
  "interface": [
    {
      "network_uuid": "string",
      "subnet_uuid": "string",
      "ip_address": "string",
      "mac_address": "string",
      "port_uuid": "string",
      "port_forward": [
        {
          "port_forwarding_type": "string",
          "port_number": "string"
        }
      ],
      "type": "string",
      "nic_id": 0,
      "anycast_ip": "string",
      "shared_ip": "string",
      "netmask": "string",
      "gateway": "string"
    }
  ]
},
"kpi_data": {
  "enabled": false,
  "kpi": [
    {
      "event_name": "string",
      "metric_value": "string",
      "metric_cond": "GT",
      "metric_type": "INT_8",
```

```
    "metric_occurrences_true": 0,
    "metric_occurrences_false": 0,
    "metric_proxy": "string",
    "metric_collector": {
      "type": "string",
      "nicid": 0,
      "proxy": "string",
      "poll_frequency": 0,
      "polling_unit": "MINUTES",
      "continuous_alarm": false
    }
  }
],
},
"rules": {
  "admin_rules": {
    "rule": [
      {
        "event_name": "string",
        "action": [
          "string"
        ]
      }
    ]
  },
  "user_rules": {
    "rule": [
      {
        "event_name": "string",
        "action": [
          "string"
        ]
      }
    ]
  }
},
"config_data": {
  "configuration": [
    {
      "dst": "string",
      "file": "string",
      "data": "string",
      "variable": [
        {
          "name": "string",
          "val": [
            "string"
          ]
        }
      ]
    }
  ]
}
```

```
    }
  ]
}
},
"port_forwarding": {
  "port": [
    {
      "type": "string",
      "protocol": "string",
      "vnf_port": 0,
      "external_port_range": [
        {
          "start": 0,
          "end": 0
        }
      ]
    }
  ]
},
"scaling": {
  "min_active": 0,
  "min_standby": 0,
  "max_active": 0,
  "max_standby": 0,
  "standby_ratio": "string",
  "elastic": false,
  "static_mac_address_pool": [
    {
      "network": "string",
      "mac_address_range": [
        {
          "start": "string",
          "end": "string"
        }
      ],
      "mac_address": [
        "string"
      ]
    }
  ],
  "static_license_pool": {
    "license": [
      {
        "license_key_value": "string",
        "license_key_type": "MAC_ADDR",
        "license_file": "string",
        "license_data": "string"
      }
    ]
  }
}
```

```
    }
  ]
},
"static_ip_address_pool": [
  {
    "network": "string",
    "gateway": "string",
    "netmask": "string",
    "ip_address_range": [
      {
        "start": "string",
        "end": "string"
      }
    ],
    "ip_address": [
      "string"
    ]
  }
],
"placement": [
  {
    "type": "AFFINITY",
    "enforcement": "STRICT",
    "host": "string",
    "zone": "string"
  }
],
"shared_ip": [
  {
    "nic_id": 0,
    "shared_ip": "string"
  }
],
"properties": {
  "property": [
    {
      "name": "string",
      "value": "string"
    }
  ]
},
"recovery_policy": {
  "action_on_recovery": "REBOOT_THEN_REDEPLOY"
},
"extensions": {
  "extension": [
    {
```

```
        "name": "string",
        "properties": {
          "property": [
            {
              "name": "string",
              "value": "string"
            }
          ]
        },
        "containers": {
          "container": [
            {
              "name": "string",
              "properties": {
                "property": [
                  {
                    "name": "string",
                    "value": "string"
                  }
                ]
              }
            }
          ]
        }
      ]
    },
    "networks": {
      "network": [
        {
          "name": "string",
          "shared": false,
          "admin_state": false,
          "router_external": false,
          "provider_physical_network": "string",
          "provider_network_type": "string",
          "provider_segmentation_id": 0,
          "subnet": [
            {
              "name": "string",
```



```
        "ipversion": "IPV_4",
        "dhcp": false,
        "address": "string",
        "netmask": "string",
        "gateway": "string"
    }
],
"switch_name": "string",
"vlan_id": 0,
"number_of_ports": 0
}
]
},
"images": {
    "image": [
        {
            "name": "string",
            "src": "string",
            "size": 0,
            "sizeunit": "MI_B",
            "disk_format": "QCOW_2",
            "container_format": "string",
            "serial_console": false,
            "e1000_net": false,
            "virtio_net": false,
            "disk_bus": "IDE"
        }
    ]
},
"flavors": {
    "flavor": [
        {
            "name": "string",
            "vcpus": 0,
            "sockets": 0,
            "cores": 0,
            "threads": 0,
            "memory_mb": 0,
            "root_disk_mb": 0,
            "ephemeral_disk_mb": 0,
            "swap_disk_mb": 0
        }
    ]
}
},
"dbDeployment": {
    "deploymentId": "string",
    "deploymentName": "string",
```

```
"tenantId": "string",
"serviceRegistrationId": "string",
"eventTime": "string",
"eventNotificationContentType": 0,
"eventNotificationType": 0,
"eventNotificationUrl": "string",
"dataModel": "string",
"classMetadata": {
  "entityName": "string",
  "propertyNames": [
    "string"
  ],
  "naturalIdentifierProperties": [
    0
  ],
  "identifierType": {
    "associationType": false,
    "entityType": false,
    "anyType": false,
    "componentType": false,
    "xmlelement": false,
    "mutable": false,
    "collectionType": false,
    "name": "string"
  },
  "identifierPropertyName": "string",
  "propertyTypes": [
    {
      "associationType": false,
      "entityType": false,
      "anyType": false,
      "componentType": false,
      "xmlelement": false,
      "mutable": false,
      "collectionType": false,
      "name": "string"
    }
  ],
  "mutable": false,
  "propertyNullability": [
    true
  ],
  "versioned": false,
  "versionProperty": 0,
  "propertyLaziness": [
    true
  ],
  "inherited": false
```

```
},
  "id": "string"
},
"deployment_details": [
  {
    "sharedIps": [
      {
        "nic_id": 0,
        "shared_ip": "string"
      }
    ],
    "vm_uuid": "string",
    "host_uuid": "string",
    "host_name": "string",
    "vm_name": "string",
    "vm_group_name": "string",
    "volume": [
      {
        "external_volume_id": "string",
        "internal_volume_id": "string",
        "size": "string",
        "sizeunit": "string",
        "bus": "string"
      }
    ],
    "interface": [
      {
        "network_uuid": "string",
        "subnet_uuid": "string",
        "ip_address": "string",
        "mac_address": "string",
        "port_uuid": "string",
        "port_forward": [
          {
            "port_forwarding_type": "string",
            "port_number": "string"
          }
        ],
        "type": "string",
        "nic_id": 0,
        "anycast_ip": "string",
        "shared_ip": "string",
        "netmask": "string",
        "gateway": "string"
      }
    ],
    "vm_state_machine_state": "string"
  }
}
```

```

],
"internal_tenant_id": "string",
"external_tenant_id": "string",
"internal_service_registration_id": "string",
"internal_deployment_id": "string",
"external_service_registration_id": "string",
"external_deployment_id": "string",
"deployment_stage": "string",
"deployment_name": "string",
"service_name": "string",
"service_version": "string",
"service_state_machine_state": "string"
}

```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>internal_deployment_id</b>	<input type="text" value="(required)"/>	<b>Internal ID of the deployment</b>	path	string

Try it out!

**POST** /v0/deployments/{internal\_deployment\_id}

Created a deployment

### Implementation Notes

Create a deployment in ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>Callback-ESC-Events</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC event notifications</b>	header	string
<b>internal_deployment_id</b>	<input type="text" value="(required)"/>	<b>Internal ID of the deployment</b>	path	string

## datamodelPayload

(required)

Parameter content type:

application/xml

ESC  
datamodel  
to deploy

body

Model | Model Schema

```
{
  "services": {
    "service_definition": [
      {
        "deployment_name": "st
        "name": "string",
        "version": "string",
        "policies": {
          "external_dependenci
            "service_dependenc
              "service": [
                {
                  "service_nam
                  "service_ver
                }
              ]
            },
          "network_dependenc
            "network": [
              "string"
            ]
          }
        },
        "placement": [
          {
            "type": "AFFINIT
            "enforcement": "
            "host": "string"
            "zone": "string"
          }
        ],
        "startup": [
          {
            "target_vm_group
            "depends_on": [
              {
                "vm_group_re
                "kpi_event_r
              }
            ]
          }
        ]
      }
    ],
    "networks": {
      "network": [
        {
          "name": "string"
          "shared": false,
          "admin_state": f
          "router_external
          "provider_physic
```

```
"provider_network": "provider_network",
"provider_segment": "provider_segment",
"subnet": [
  {
    "name": "string",
    "ipversion": "string",
    "dhcp": false,
    "address": "string",
    "netmask": "string",
    "gateway": "string"
  }
],
"switch_name": "string",
"vlan_id": 0,
"number_of_ports": 0
}
],
},
"volumes": {
  "volume": [
    {
      "external_volume": "string",
      "internal_volume": "string",
      "size": "string",
      "sizeunit": "string",
      "bus": "string"
    }
  ]
},
"vm_group": [
  {
    "name": "string",
    "image": "string",
    "flavor": "string",
    "bootup_time": 0,
    "recovery_wait_time": 0,
    "misc": {
      "watchdog": {
        "hardware": "string",
        "action": "string"
      }
    }
  }
],
"volumes": {
  "volume": [
    {
      "external_volume": "string",
      "internal_volume": "string",
      "size": "string",
      "sizeunit": "string",
      "bus": "string"
    }
  ]
},
"interfaces": {
```

```
"interface": [
  {
    "network_uid": "network_uid",
    "subnet_uid": "subnet_uid",
    "ip_address": "ip_address",
    "mac_address": "mac_address",
    "port_uuid": "port_uuid",
    "port_forwarding": {
      "port_forwarding": "port_forwarding",
      "port_number": "port_number"
    }
  },
  {
    "type": "string",
    "nic_id": 0,
    "anycast_ip": "anycast_ip",
    "shared_ip": "shared_ip",
    "netmask": "netmask",
    "gateway": "gateway"
  }
],
},
"kpi_data": {
  "enabled": false,
  "kpi": [
    {
      "event_name": "event_name",
      "metric_value": "metric_value",
      "metric_condition": "metric_condition",
      "metric_type": "metric_type",
      "metric_occurrence": "metric_occurrence",
      "metric_proxy": "metric_proxy",
      "metric_collection": "metric_collection",
      "type": "string",
      "nic_id": 0,
      "proxy": "proxy",
      "poll_frequency": "poll_frequency",
      "polling_unit": "polling_unit",
      "continuous": "continuous"
    }
  ]
},
},
"rules": {
  "admin_rules": {
    "rule": [
      {
        "event_name": "event_name",
        "action": "string"
      }
    ]
  }
}
]
```

```
    },
    "user_rules": {
      "rule": [
        {
          "event_name": "string"
          "action": "string"
        }
      ]
    }
  },
  "config_data": {
    "configuration": {
      {
        "dst": "string"
        "file": "string"
        "data": "string"
        "variable": {
          {
            "name": "string"
            "val": [
              "string"
            ]
          }
        ]
      }
    ]
  },
  "port_forwarding": {
    "port": [
      {
        "type": "string"
        "protocol": "string"
        "vnf_port": "string"
        "external_port": {
          {
            "start": "string"
            "end": "string"
          }
        ]
      }
    ]
  },
  "scaling": {
    "min_active": 0,
    "min_standby": 0,
    "max_active": 0,
    "max_standby": 0,
    "standby_ratio": "string"
    "elastic": false
    "static_mac_address": {
      "network": "string"
    }
  }
}
```



```
        "mac_address": {
            {
                "start": "
                "end": "
            }
        ],
        "mac_address": {
            "string"
        }
    ],
    "static_license": {
        "license": [
            {
                "license_k":
                "license_k":
                "license_f":
                "license_d":
            }
        ]
    },
    "static_ip_address": {
        {
            "network": "
            "gateway": "
            "netmask": "
            "ip_address": {
                {
                    "start": "
                    "end": "
                }
            },
            "ip_address": {
                "string"
            }
        }
    ],
    "placement": [
        {
            "type": "AFFIN
            "enforcement":
            "host": "strin
            "zone": "strin
        }
    ],
    "shared_ip": [
        {
            "nic_id": 0,
            "shared_ip": "
        }
    ],
    "properties": {
        "property": [
```

```
        {
          "name": "str
          "value": "st
        }
      ]
    },
    "recovery_policy":
      "action_on_recov
    },
    "extensions": {
      "extension": [
        {
          "name": "str
          "properties"
          "property"
            {
              "name"
              "value
            }
          ]
        },
        "containers"
          "container
            {
              "name"
              "prope
              "pro
                {
                }
              ]
            }
          ]
        }
      ]
    },
    "tenants": {
      "tenant": [
        {
          "name": "string",
          "networks": {
            "network": [
              {
                "name": "string"
                "shared": false,
                "admin_state": f
```

```
"router_external
"provider_physic
"provider_networ
"provider_segmen
"subnet": [
  {
    "name": "str
    "ipversion":
    "dhcp": fals
    "address": "
    "netmask": "
    "gateway": "
  }
],
"switch_name": "
"vlan_id": 0,
"number_of_ports
}
]
},
"services": {
  "service_definition"
  {
    "deployment_name
    "name": "string"
    "version": "stri
    "policies": {
      "external_depe
      "service_dep
      "service":
        {
          "servi
          "servi
        }
      ]
    },
    "network_dep
    "network":
      "string"
    ]
  }
},
"placement": [
  {
    "type": "A
    "enforceme
    "host": "s
    "zone": "s
  }
],
"startup": [
  {
    "target_vm
    "depends_o
```

```
        {
            "vm_group": "vm_group",
            "kpi_endpoint": "kpi_endpoint"
        }
    ]
}
],
},
"networks": {
    "network": [
        {
            "name": "shared",
            "shared": true,
            "admin_state": "admin_state",
            "router_external": "router_external",
            "provider_network": "provider_network",
            "provider_network": "provider_network",
            "provider_network": "provider_network",
            "subnet": {
                "name": "shared",
                "ip_version": "ip_version",
                "dhcp_enabled": "dhcp_enabled",
                "address": "address",
                "netmask": "netmask",
                "gateway": "gateway"
            }
        },
        {
            "name": "shared",
            "shared": true,
            "admin_state": "admin_state",
            "router_external": "router_external",
            "provider_network": "provider_network",
            "provider_network": "provider_network",
            "provider_network": "provider_network",
            "subnet": {
                "name": "shared",
                "ip_version": "ip_version",
                "dhcp_enabled": "dhcp_enabled",
                "address": "address",
                "netmask": "netmask",
                "gateway": "gateway"
            }
        }
    ],
    "switch_name": "switch_name",
    "vlan_id": "vlan_id",
    "number_of_ports": "number_of_ports"
}
],
},
"volumes": {
    "volume": [
        {
            "external_id": "external_id",
            "internal_id": "internal_id",
            "size": "size",
            "sizeunit": "sizeunit",
            "bus": "storage"
        }
    ]
}
],
},
"vm_group": [
    {
        "name": "storage",
        "image": "storage",
        "flavor": "storage",
        "bootup_time": "bootup_time",
        "recovery_wait": "recovery_wait",
        "misc": {
            "watchdog": "watchdog"
        }
    }
]
```

```
        "hardwar
        "action"
    }
},
"volumes": {
    "volume":
    {
        "exter
        "inter
        "size"
        "sizeu
        "bus":
    }
}
],
},
"interfaces"
"interface
{
    "netwo
    "subne
    "ip_ad
    "mac_a
    "port_
    "port_
    {
        "p
        "p
    }
},
    "type"
    "nic_i
    "anyca
    "share
    "netma
    "gatew
}
]
},
"kpi_data":
    "enabled":
    "kpi": [
    {
        "event
        "metri
        "metri
        "metri
        "metri
        "metri
        "metri
        "typ
        "nic
        "pro
        "pol
```

```
        "pol
        "con
    }
    ]
},
"rules": {
    "admin_rul
    "rule":
    {
        "eve
        "act
        "s
    ]
    }
    ]
},
"user_rule
"rule":
    {
        "eve
        "act
        "s
    ]
    }
    ]
},
"config_data
"configura
    {
        "dst":
        "file"
        "data"
        "varia
        {
            "n
            "v
        ]
    }
    ]
},
"port_forwan
"port": [
    {
        "type"
        "proto
        "vnf_p
        "exter
        {
            "s
```

```
        "e
        }
    ]
}
},
"scaling": {
    "min_activ
    "min_stand
    "max_activ
    "max_stand
    "standby_r
    "elastic":
    "static_ma
    {
        "netwo
        "mac_a
        {
            "s
            "e
        }
    },
    "mac_a
    "str
    ]
}
},
"static_li
    "license
    {
        "lic
        "lic
        "lic
        "lic
    }
}
},
"static_ip.
    {
        "netwo
        "gatew
        "netma
        "ip_ad
        {
            "s
            "e
        }
    },
    "ip_ad
    "str
    ]
}
}
},
```

```
"placement":
  {
    "type":
    "enforce
    "host":
    "zone":
  }
],
"shared_ip":
  {
    "nic_id"
    "shared_
  }
],
"properties"
  "property"
  {
    "name"
    "value
  }
]
},
"recovery_po
  "action_on
},
"extensions"
  "extension
  {
    "name"
    "prope
    "pro
    {
  }
]
},
"conta
  "con
  {
}
]
}
```



```
    }
  }
]
},
"deployments": {
  "deployment": [
    {
      "name": "string"
      "policies": {
        "external_depe
        "service_dep
        "service":
          {
            "servi
            "servi
          }
        ]
      },
      "network_dep
      "network":
        "string"
      ]
    },
    "placement": [
      {
        "type": "A
        "enforceme
        "host": "s
        "zone": "s
      }
    ],
    "startup": [
      {
        "target_vm
        "depends_o
          {
            "vm_gr
            "kpi_e
          }
        ]
      }
    ],
    "networks": {
      "network": [
        {
          "name": "s
          "shared":
          "admin_sta
          "router_ex
          "provider_
```

```
"provider_
"provider_
"subnet":
  {
    "name"
    "ipver
    "dhcp"
    "addre
    "netma
    "gatew
  }
],
"switch_na
"vlan_id":
"number_of
}
]
},
"volumes": {
  "volume": [
    {
      "external_
      "internal_
      "size": "s
      "sizeunit"
      "bus": "st
    }
  ]
},
"vm_group": [
  {
    "name": "str
    "image": "st
    "flavor": "s
    "bootup_time
    "recovery_wa
    "misc": {
      "watchdog"
      "hardwar
      "action"
    }
  },
  "volumes": {
    "volume":
      {
        "exter
        "inter
        "size"
        "sizeu
        "bus":
      }
    ]
  },
  "interfaces"
```

```
"interface
  {
    "netwo
    "subne
    "ip_ad
    "mac_a
    "port_
    "port_
    {
      "p
      "p
    }
  },
  "type"
  "nic_i
  "anyca
  "share
  "netma
  "gatew
}
]
},
"kpi_data":
  "enabled":
    "kpi": [
      {
        "event
        "metri
        "metri
        "metri
        "metri
        "metri
        "metri
        "metri
        "typ
        "nic
        "pro
        "pol
        "pol
        "con
      }
    ]
  },
"rules": {
  "admin_rul
  "rule":
    {
      "eve
      "act
      "s
    ]
  }
}
```

```
    },
    "user_rule":
      "rule":
        {
          "eve":
            "act":
              "s"
            ]
          }
        ]
      },
    "config_data":
      "configura":
        {
          "dst":
            "file":
            "data":
            "varia":
              {
                "n":
                "v":
                ]
              }
            ]
          }
        ]
      },
    "port_forwar":
      "port": [
        {
          "type":
          "proto":
          "vnf_p":
          "exter":
            {
              "s":
              "e":
            }
          ]
        }
      ]
    },
    "scaling": {
      "min_activ":
      "min_stand":
      "max_activ":
      "max_stand":
      "standby_r":
      "elastic":
      "static_ma":
        {
          "netwo
```

```
        "mac_a
        {
            "s
            "e
        }
    ],
    "mac_a
        "str
    ]
}
],
"static_li
"license
{
    "lic
    "lic
    "lic
    "lic
}
]
},
"static_ip.
{
    "netwo
    "gatew
    "netma
    "ip_ad
    {
        "s
        "e
    }
    ],
    "ip_ad
        "str
    ]
}
],
},
"placement":
{
    "type":
    "enforce
    "host":
    "zone":
}
],
"shared_ip":
{
    "nic_id"
    "shared_
}
],
"properties"
"property"
```

```
        {
            "name"
            "value
        }
    ]
},
"recovery_po
"action_on
},
"extensions"
"extension
{
    {
        "name"
        "prope
        "pro
        {
            }
        ]
    },
    "conta
    "con
    {
        }
    ]
},
"networks": {
    "network": [
        {
            "name": "string",
            "shared": false,
            "admin_state": false,
            "router_external": fal
```

```

"provider_physical_net
"provider_network_type
"provider_segmentation
"subnet": [
  {
    "name": "string",
    "ipversion": "IPV_
    "dhcp": false,
    "address": "string
    "netmask": "string
    "gateway": "string
  }
],
"switch_name": "string
"vlan_id": 0,
"number_of_ports": 0
}
]
},
"images": {
  "image": [
    {
      "name": "string",
      "src": "string",
      "size": 0,
      "sizeunit": "MI_B",
      "disk_format": "QCOW_2
      "container_format": "s
      "serial_console": fals
      "e1000_net": false,
      "virtio_net": false,
      "disk_bus": "IDE"
    }
  ]
},
"flavors": {
  "flavor": [
    {
      "name": "string",
      "vcpus": 0,
      "sockets": 0,
      "cores": 0,
      "threads": 0,
      "memory_mb": 0,
      "root_disk_mb": 0,
      "ephemeral_disk_mb": 0
      "swap_disk_mb": 0
    }
  ]
}
}
}

```

Click to set as parameter value

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
default	successful operation		

[Try it out!](#)

**PUT** /v0/deployments/{internal\_deployment\_id} Update an existing deployment in ESC

### Implementation Notes

Update an existing deployment in ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>Callback-ESC-Events</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC event notifications</b>	header	string
<b>internal_deployment_id</b>	<input type="text" value="(required)"/>	<b>Internal ID of the deployment</b>	path	string
body	<div style="border: 1px solid #ccc; height: 100px; width: 100%;"></div> <p>Parameter content type: <input type="text" value="application/xml"/></p>		body	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
default	successful operation		

[Try it out!](#)

## Flavors



GET

/v0/flavors

Get all flavors

## Implementation Notes

All flavors created in ESC will be returned as a List

## Response Class (Status 200)

Model **Model Schema**

```
[
  {
    "name": "string",
    "vcpus": 0,
    "memory_mb": 0,
    "root_disk_mb": 0,
    "ephemeral_disk_mb": 0,
    "swap_disk_mb": 0,
    "service_registration_id": "string",
    "event_type": "string",
    "external_flavor_id": "string",
    "internal_flavor_id": "string",
    "public": false
  }
]
```

Response Content Type [Try it out!](#)

DELETE

/v0/flavors/{internalFlavorId}

Delete a flavor from ESC

## Implementation Notes

Delete a flavor from ESC

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalFlavorId</b>	<input type="text" value="(required)"/>	<b>Internal flavor ID</b>	path	string

## Response Messages

HTTP Status

Code	Reason	Response Model	Headers
default	successful operation		

Try it out!

**GET** /v0/flavors/{internalFlavorId} Get the flavor by the internal flavor ID

### Implementation Notes

Flavors not created by ESC will not be checked

### Response Class (Status 200)

Model | Model Schema

```

{
  "name": "string",
  "vcpus": 0,
  "memory_mb": 0,
  "root_disk_mb": 0,
  "ephemeral_disk_mb": 0,
  "swap_disk_mb": 0,
  "service_registration_id": "string",
  "event_type": "string",
  "external_flavor_id": "string",
  "internal_flavor_id": "string",
  "public": false
}

```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>internalFlavorId</b>	<input type="text" value="(required)"/>	<b>Internal flavor ID</b>	path	string

Try it out!

**POST** /v0/flavors/{internalFlavorId} Create a flavor in ESC

### Implementation Notes

Create a flavor in ESC

## Response Class (Status 200)

Model | Model Schema

```
{
  "name": "string",
  "vcpus": 0,
  "memory_mb": 0,
  "root_disk_mb": 0,
  "ephemeral_disk_mb": 0,
  "swap_disk_mb": 0,
  "service_registration_id": "string",
  "event_type": "string",
  "external_flavor_id": "string",
  "internal_flavor_id": "string",
  "public": false
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalFlavorId</b>	<input type="text" value="(required)"/>	<b>Internal flavor ID</b>	path	string
<b>body</b>	<div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;"><p>(required)</p></div> <p>Parameter content type: <input type="text" value="application/xml"/></p>	<b>ESC datamodel with the flavor to create</b>	body	string

Try it out!

## Images

GET /v0/images [Get all images](#)

### Implementation Notes

All images created in ESC will be returned as a List

## Response Class (Status 200)

Model | Model Schema

```
[
  {
    "serviceRegistrationId": "string",
    "imageChecksum": "string",
    "imageIsEnabled": false,
    "external_image_id": "string",
    "external_tenant_id": "string",
    "name": "string",
    "src": "string",
    "disk_format": "string",
    "container_format": "string",
    "e1000_net": false,
    "virtio_net": false,
    "disk_bus": "string",
    "size": 0,
    "sizeunit": "string",
    "event_type": "string",
    "internal_tenant_id": "string",
    "public": false,
    "internal_image_id": "string",
    "serial_console": false,
    "cloud_init_support": "string"
  }
]
```

Response Content Type

[Try it out!](#)

DELETE

/v0/images/{internalImageId}

Delete an image from ESC

### Implementation Notes

Delete an image from ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalImageId</b>	<input type="text" value="(required)"/>	<b>Internal image ID</b>	path	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
default	successful operation		

Try it out!

GET /v0/images/{internalImageId}

Get the image by the internal image ID

### Implementation Notes

Images not created by ESC will not be checked

### Response Class (Status 200)

Model | Model Schema

```
{
  "serviceRegistrationId": "string",
  "imageChecksum": "string",
  "imageIsEnabled": false,
  "external_image_id": "string",
  "external_tenant_id": "string",
  "name": "string",
  "src": "string",
  "disk_format": "string",
  "container_format": "string",
  "e1000_net": false,
  "virtio_net": false,
  "disk_bus": "string",
  "size": 0,
  "sizeunit": "string",
  "event_type": "string",
  "internal_tenant_id": "string",
  "public": false,
  "internal_image_id": "string",
  "serial_console": false,
  "cloud_init_support": "string"
}
```

Response Content Type

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>internalImageId</b>	<input type="text" value="(required)"/>	Internal image ID	path	string

Try it out!

POST /v0/images/{internalImageId}

Create an image in ESC

## Implementation Notes

Create an image in ESC

## Response Class (Status 200)

Model | Model Schema

```
{
  "serviceRegistrationId": "string",
  "imageChecksum": "string",
  "imageIsEnabled": false,
  "external_image_id": "string",
  "external_tenant_id": "string",
  "name": "string",
  "src": "string",
  "disk_format": "string",
  "container_format": "string",
  "e1000_net": false,
  "virtio_net": false,
  "disk_bus": "string",
  "size": 0,
  "sizeunit": "string",
  "event_type": "string",
  "internal_tenant_id": "string",
  "public": false,
  "internal_image_id": "string",
  "serial_console": false,
  "cloud_init_support": "string"
}
```

Response Content Type

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalImageId</b>	<input type="text" value="(required)"/>	<b>Internal image ID</b>	path	string
<b>body</b>		<b>ESC datamodel with</b>	body	string

the image to create

(required)

Parameter content type:

application/xml

Try it out!

## Networks

GET /v0/{internalTenantId}/networks

Get all networks under a tenant

### Implementation Notes

All networks under that tenant in ESC will be returned as a List

### Response Class (Status 200)

Model | Model Schema

```
[
  {
    "status": "string",
    "name": "string",
    "shared": false,
    "network_id": "string",
    "tenant_id": "string",
    "external_tenant_id": "string",
    "external_network_id": "string",
    "subnet": [
      "string"
    ],
    "admin_state_up": false,
    "event_type": "string",
    "internal_tenant_id": "string",
    "internal_network_id": "string",
    "provider_segmentation_id": "string",
    "router_external": false,
    "provider_physical_network": "string",
    "provider_network_type": "string",
    "vlan_id": 0,
    "switch_name": "string",
    "number_of_ports": 0
  }
]
```

```
}  
]
```

Response Content Type

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the network</b>	path	string

Try it out!

DELETE

/v0/{internalTenantId}/networks/{internalNetworkId}

Delete a network from an existing tenant in ESC

## Implementation Notes

Delete a network from ESC

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalNetworkId</b>	<input type="text" value="(required)"/>	<b>Internal network ID</b>	path	string
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the network</b>	path	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
default	successful operation		

Try it out!

GET

/v0/{internalTenantId}/networks/{internalNetworkId}

Get a network by its tenant and internal network ID

## Implementation Notes



Only networks created using ESC will be checked

## Response Class (Status 200)

Model | Model Schema

```
{
  "status": "string",
  "name": "string",
  "shared": false,
  "network_id": "string",
  "tenant_id": "string",
  "external_tenant_id": "string",
  "external_network_id": "string",
  "subnet": [
    "string"
  ],
  "admin_state_up": false,
  "event_type": "string",
  "internal_tenant_id": "string",
  "internal_network_id": "string",
  "provider_segmentation_id": "string",
  "router_external": false,
  "provider_physical_network": "string",
  "provider_network_type": "string",
  "vlan_id": 0,
  "switch_name": "string",
  "number_of_ports": 0
}
```

Response Content Type

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the network</b>	path	string
<b>internalNetworkId</b>	<input type="text" value="(required)"/>	<b>Internal network ID</b>	path	string

Try it out!

**POST** /v0/{internalTenantId}/networks/{internalNetworkId}

Create a network under an existing tenant in ESC

## Implementation Notes

Create a network in ESC

## Response Class (Status 200)

Model | Model Schema

```
{
  "status": "string",
  "name": "string",
  "shared": false,
  "network_id": "string",
  "tenant_id": "string",
  "external_tenant_id": "string",
  "external_network_id": "string",
  "subnet": [
    "string"
  ],
  "admin_state_up": false,
  "event_type": "string",
  "internal_tenant_id": "string",
  "internal_network_id": "string",
  "provider_segmentation_id": "string",
  "router_external": false,
  "provider_physical_network": "string",
  "provider_network_type": "string",
  "vlan_id": 0,
  "switch_name": "string",
  "number_of_ports": 0
}
```

Response Content Type

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the network</b>	path	string
<b>internalNetworkId</b>	<input type="text" value="(required)"/>	<b>Internal network ID</b>	path	string

**body**

(required)

Parameter content type:

application/xml

**Network configuration payload**

body

Model | Model Schema

```

{
  "status": "string",
  "name": "string",
  "shared": false,
  "network_id": "string",
  "tenant_id": "string",
  "external_tenant_id": "string",
  "external_network_id": "string",
  "subnet": [
    "string"
  ],
  "admin_state_up": false,
  "event_type": "string",
  "internal_tenant_id": "string",
  "internal_network_id": "string",
  "provider_segmentation_id": "string",
  "router_external": false,
  "provider_physical_network": "string",
  "provider_network_type": "string",
  "vlan_id": 0,
  "switch_name": "string",
  "number_of_ports": 0
}

```

Click to set as parameter value

Try it out!

POST
/v0/{internalTenantId}/networks/{internalNetworkId}/add

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
Client-Transaction-Id	<input type="text"/>		header	string
<b>internalTenantId</b>	<input type="text" value="(required)"/>		path	string
<b>internalNetworkId</b>	<input type="text" value="(required)"/>		path	string

**body**

Parameter content

**body**

Model | Model Schema

```

{
  "status": "string",
  "name": "string",
  "shared": false,
  "network id": "string",

```

Parameter content type:

application/json

```
"tenant_id": "string",  
  
"external_tenant_id": "string",  
"external_network_id": "string",  
"subnet": [  
  "string"  
],  
"admin_state_up": false,  
"event_type": "string",  
"internal_tenant_id": "string",  
"internal_network_id": "string",  
"provider_segmentation_id": "string",  
"router_external": false,  
"provider_physical_network": "string",  
"provider_network_type": "string",  
"vlan_id": 0,  
"switch_name": "string",  
"number_of_ports": 0  
}
```

Click to set as parameter value

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
default	successful operation		

Try it out!

**POST** /v0/{internalTenantId}/networks/{internalNetworkId}/remove

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
Client-Transaction-Id	<input type="text"/>		header	string
<b>internalTenantId</b>	<input type="text" value="(required)"/>		path	string
<b>internalNetworkId</b>	<input type="text" value="(required)"/>		path	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
default	successful operation		

Try it out!

# Subnets

GET /v0/{internalTenantId}/subnets

Get all subnets under a network

## Implementation Notes

All subnet under that network in ESC will be returned as a List

## Response Class (Status 200)

Model | Model Schema

```
[
  {
    "external_tenant_id": "string",
    "external_network_id": "string",
    "external_subnet_id": "string",
    "name": "string",
    "allocation_pools": [
      {
        "start": "string",
        "end": "string"
      }
    ],
    "gateway_ip": "string",
    "ip_version": 0,
    "cidr": "string",
    "enable_dhcp": false,
    "network_id": "string",
    "tenant_id": "string",
    "event_type": "string",
    "internal_tenant_id": "string",
    "subnet_id": "string",
    "internal_network_id": "string",
    "internal_subnet_id": "string"
  }
]
```

Response Content Type

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
internalNetworkId	<input type="text"/>	Internal network ID of the subnet	header	string
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the subnet</b>	path	string

Try it out!

**DELETE** /v0/{internalTenantId}/subnets/{internalSubnetId}

Delete a subnet from an existing tenant in ESC

### Implementation Notes

Delete a subnet from ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the subnet</b>	path	string
<b>internalSubnetId</b>	<input type="text" value="(required)"/>	<b>Internal subnet ID</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
default	successful operation		

Try it out!

**GET** /v0/{internalTenantId}/subnets/{internalSubnetId}

Get a subnet by its tenant and internal subnet ID

### Implementation Notes


Only subnets created using ESC will be checked

### Response Class (Status 200)

Model | Model Schema

```
{
  "external_tenant_id": "string",
  "external_network_id": "string",
  "external_subnet_id": "string",
  "name": "string",
}
```

```
"allocation_pools": [
  {
    "start": "string",
    "end": "string"
  }
],
"gateway_ip": "string",
"ip_version": 0,
"cidr": "string",
"enable_dhcp": false,
"network_id": "string",
"tenant_id": "string",
"event_type": "string",
"internal_tenant_id": "string",
"subnet_id": "string",
"internal_network_id": "string",
"internal_subnet_id": "string"
}
```

Response Content Type  

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the subnet</b>	path	string
<b>internalSubnetId</b>	<input type="text" value="(required)"/>	<b>Internal subnet ID</b>	path	string

[Try it out!](#)

**POST** /v0/{internalTenantId}/subnets/{internalSubnetId}

Create a subnet under an existing tenant in ESC

## Implementation Notes

Create a subnet in ESC

## Response Class (Status 200)

Model [Model Schema](#)

```
{
  "external_tenant_id": "string",
  "external_network_id": "string",
  "external_subnet_id": "string",
}
```

```

"name": "string",
"allocation_pools": [
  {
    "start": "string",
    "end": "string"
  }
],
"gateway_ip": "string",
"ip_version": 0,
"cidr": "string",
"enable_dhcp": false,
"network_id": "string",
"tenant_id": "string",
"event_type": "string",
"internal_tenant_id": "string",
"subnet_id": "string",
"internal_network_id": "string",
"internal_subnet_id": "string"
}

```

Response Content Type

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the subnet</b>	path	string
<b>internalSubnetId</b>	<input type="text" value="(required)"/>	<b>Internal subnet ID</b>	path	string
<b>body</b>	<input type="text" value="(required)"/> Parameter content type: <input type="text" value="application/xml"/>	<b>Subnet configuration payload</b>	body	Model   Model Schema <pre> {   "external_tenant_id": "string",   "external_network_id": "string",   "external_subnet_id": "string",   "name": "string",   "allocation_pools": [     {       "start": "string",       "end": "string"     }   ] } </pre>



```
],
"gateway_ip": "string",
"ip_version": 0,
"cidr": "string",
"enable_dhcp": false,
"network_id": "string",
"tenant_id": "string",
"event_type": "string",
"internal_tenant_id": "string",
"subnet_id": "string",
"internal_network_id": "string",
"internal_subnet_id": "string"
}
```

Click to set as parameter value

Try it out!

## Tenants

GET /v0/tenants

Get all tenants

### Implementation Notes

All tenants in ESC will be returned as a List

### Response Class (Status 200)

Model | Model Schema

```
[
  {
    "external_tenant_id": "string",
    "name": "string",
    "enabled": false,
    "description": "string",
    "tenant_id": "string",
    "event_type": "string",
    "internal_tenant_id": "string"
  }
]
```

Response Content Type application/xml

Try it out!

DELETE /v0/tenants/{internal\_tenant\_id}

Delete a tenant from ESC

## Implementation Notes

Delete a tenant from ESC

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>internal_tenant_id</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID</b>	path	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
default	successful operation		

[Try it out!](#)

**GET** /v0/tenants/{internal\_tenant\_id}

Get a tenant by its name

## Implementation Notes

Only tenants created using ESC will be checked

## Response Class (Status 200)

Model | **Model Schema**

```
{
  "external_tenant_id": "string",
  "name": "string",
  "enabled": false,
  "description": "string",
  "tenant_id": "string",
  "event_type": "string",
  "internal_tenant_id": "string"
}
```

Response Content Type

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>internal_tenant_id</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID</b>	path	string

[Try it out!](#)

POST

/v0/tenants/{internal\_tenant\_id}

Create a tenant in ESC

## Implementation Notes

Create a tenant in ESC

## Response Class (Status 200)

Model | Model Schema

```
{
  "external_tenant_id": "string",
  "name": "string",
  "enabled": false,
  "description": "string",
  "tenant_id": "string",
  "event_type": "string",
  "internal_tenant_id": "string"
}
```

Response Content Type

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>body</b>	<input type="text" value="(required)"/> Parameter content type: <input type="text" value="application/xml"/>	<b>Tenant configuration payload</b>	body	Model   Model Schema <pre>{   "external_tenant_id": "string",   "name": "string",   "enabled": false,   "description": "string",   "tenant_id": "string",   "event_type": "string",   "internal_tenant_id": "string" }</pre> Click to set as parameter value
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internal_tenant_id</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID</b>	path	string

Try it out!

# VM Operations

POST

/v0/{internal\_tenant\_id}/deployments/vm/{vm\_name}

Handle VM operation request with the operation type as payload

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for notifications</b>	header	string
<b>Callback-ESC-Events</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC event notifications</b>	header	string
<b>internal_tenant_id</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID</b>	path	string
<b>vm_name</b>	<input type="text" value="(required)"/>	<b>VM name</b>	path	string
<b>body</b>	<div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;"><input type="text" value="(required)"/></div> <p>Parameter content type: <input type="text" value="application/xml"/></p>	<b>VM operation payload specifying the type of operation</b>	body	<p>Model   Model Schema</p> <pre>{   "operation": "string" }</pre> <p>Click to set as parameter value</p>

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	VM operation request accepted		
400	Missing callback		
404	VM or tenant not found		

Try it out!