Lenovo XClarity Orchestrator Provider for Terraform User's Guide

Version 2.0.0

Note

Before using this information and the product it supports, read the general and legal notices in the XClarity Orchestrator online documentation.

First Edition (March 2023)

Contents

Summary of changes in the XClarity Orchestrator provider for Terraform

Lenovo XClarity Orchestrator supports enhancements to the XClarity Orchestrator provider for Terraform.

Version 2.0.0

There are no changes to the XClarity Orchestrator provider for Terraform in this release.

Chapter 1. Terraform integration

Terraform is an open-source *infrastructure as code* software tool for changing, configuring, and automating infrastructure (resources) using a set of predefined declarative definitions. The Lenovo XClarity Orchestrator provider is a plugin for Terraform that you can use to automatically monitor, manage, and provision resources that are managed by XClarity Orchestrator.

You can use Terraform configurations to perform the following functions.

- **Managing resource managers**. Retrieve information about, connect, and disconnect resource managers.
- **Managing servers**. Retrieve information about and modify the power state of servers.
- **Provisioning updates**. Create, modify, and , assign update-compliance policies, and apply updates to one or more resources.
- **Provision server configuration**. Create, assign and deploy configuration settings on managed servers to comply with a defined server-configuration pattern.

Chapter 2. Installing and initializing the XClarity Orchestrator provider in Terraform

To use the Lenovo XClarity Orchestrator provider commands and APIs, you must install the XClarity Orchestrator provider and initialize Terraform.

Before you begin

Ensure that Terraform v0.13 is installed.

Ensure that XClarity Orchestrator v1.3 or later is installed.

Procedure

To install the XClarity Orchestrator provider for Terraform, complete one of the following procedures.

- **Linux**
  1. Download and install Terraform from the Download Terraform webpage.
  2. Download the XClarity Orchestrator provider binary from the XClarity Orchestrator download webpage.

     Tip: The binary file is named `terraform-provider-lxco_<version>_linux_amd64`, where <*version*> is the provider version. Do not change the file name.
  3. Create the appropriate subdirectory within the user plugins directory for the XClarity Orchestrator provider.
     ```
     $ mkdir ~/.local/share/terraform/plugins/lenovo.com/xclarity/lxco/$<version>/linux_amd64
     ```
  4. Move the XClarity Orchestrator provider binary to the subdirectory that you just created in the user plugins directory.
     ```
     $ mv terraform-provider-lxco_<version>_linux_amd64
     ~/.local/share/terraform/plugins/lenovo.com/xclarity/lxco/<version>/linux_amd64
     ```
  5. Initialize the workspace to refresh the XClarity Orchestrator provider.
     ```
     $ terraform init
     ```
- **Windows**
  1. Download and install Terraform from the Download Terraform webpage.
  2. Download the XClarity Orchestrator provider binary from the XClarity Orchestrator download webpage.

     Tip: The binary file is named `terraform-provider-lxco_{version}_windows_amd64.exe`, where *{version}* is the provider version. Do not change the file name.
  3. Create the appropriate subdirectory within the user plugins directory for the XClarity Orchestrator provider.
     ```
     mkdir -p %APPDATA%\terraform.d\plugins\lenovo.com\xclarity\lxco\<version>\windows_amd64
     ```
  4. Move the XClarity Orchestrator provider binary to the subdirectory that you just created in the user plugins directory.
     ```
     move terraform-provider-lxco_<version>_windows_amd64.exe
     %APPDATA%\terraform.d\plugins\lenovo.com\xclarity\lxco\<version>\windows_amd64
     ```
  5. Initialize the workspace to refresh the XClarity Orchestrator provider.
     ```
     terraform init
     ```

Chapter 3. Using the XClarity Orchestrator provider for Terraform

Using the Terraform

To use the XClarity Orchestrator provider for Terraform, you must declare the provider in the Terraform configuration using the inputs

```
terraform {
   required_providers {
      lxco = {
         source = "lenovo.com/xclarity/lxco"
         version = ""
      }
   }
}

provider "lxco" {
   host = ""
   username = ""
   password = ""
}
```

For example:

```
terraform {
   required_providers {
      lxco = {
         source = "lenovo.com/xclarity/lxco"
         version = "0.1"
      }
   }
}

provider "lxco" {
   host = "192.0.2.0"
   username = "lxco_admin"
   password = "*********"
}
```

For more information about using Terraform and the Terraform configuration language, see the HashiCorp Terraform website.

Specifying credential

Credentials can be specified in the following ways.

- Use credentials from the XClarity Orchestrator’s security vault.
- Create **TF_VAR_username** and **TF_VAR_password** environment variables to specify your credentials. Terraform searches the environment of its own process for environment variables named “TF_VAR_” followed by the name of a declared variable and then matches the variable name exactly as given in configuration.
  1. In the main.tf file, comment out the declared variables for the credentials.
     ```
     provider "lxco" {
        host = "192.0.2.0"
        # username = ""
        # password = ""
     ```

```
}
```

2. From the Terraform console, create the environment variables.

```
$ terraform init
$ export TF_VAR_username=LXCA_USER
$ export TF_VAR_password=*password*

$ terraform plan
provider.lxco.password
  Enter a value: *password*
provider.lxco.username
  Enter a value: LXCA_USER

$ terraform apply
provider.lxco.password
  Enter a value: *password*
provider.lxco.username
  Enter a value: LXCA_USER
```

Chapter 4. Managing resource managers

You can use Terraform configurations to retrieve information about resource managers, and to connect and disconnect resource managers.

Retrieving a list of all resource managers

Use this definition to retrieve a list of all Lenovo XClarity Administrator resource manager.

Usage

```
data "lxco_manager" "all" {}

output "all_managers" {
   value = data.lxco_manager.all
}
```

Example

The following example retrieves a list of all resource managers.

```
terraform {
   required_providers {
      lxco = {
         source = "lenovo.com/xclarity/lxco"
         version = "0.1"
      }
   }
}

provider "lxco" {
   host = "192.0.2.0"
   # username = ""
   # password = ""
}

# Fetch details for all resource managers
data "lxco_manager" "all" {}

# Return details for all resource managers
output "all_managers" {
   value = data.lxco_manager.all
}
```

Inputs

None

Outputs

Name	Type	Description
auth_type	String	Authentication type. This is always **basic**.
hostname	String	Resource manager host name
password	String	Password to use for basic authentication
port	Integer	Resource manager port

Name	Type	Description
type	String	Type of resource manager. This can be one of the following values. • **Schneider EcoStruxure IT** • **vRealize Operations Manager** • **XClarity Administrator**
username	String	User name to use for basic authentication
uuid	String	Resource manager UUID

The following example is returned if the request is successful.

```
[{
   "auth_type": "basic",
   "hostname": "109.0.2.10,"
   "port": 443,
   "password": "password",
   "type": "XClarity Administrator",
   "username": "userid",
   "uuid": "8D735FCEFBCD49118C68169312166C68"
}]
```

Retrieving information about a single resource manager

Uses this definition to retrieves information about a specific Lenovo XClarity Administrator resource manager.

Usage

```
resource "lxco_manager" "manager" {
   uuid = string
}
output "manager" {
   value = data.lxco_manager.manager
}
```

Example

The following example retrieves information about a specific resource manager.

```
terraform {
   required_providers {
      lxco = {
         source = "lenovo.com/xclarity/lxco"
         version = "0.1"
      }
   }
}

provider "lxco" {
   host = "192.0.2.0"
   # username = ""
   # password = ""
}

# Fetch details for a specific resource manager
resource "lxco_manager" "manager" {
   uuid = "00632D78DE644E23B712E200FE449787"
}

# Return details for the resource manager
output "manager" {
    value = data.lxco_manager.manager
}
```

Inputs

Name	Re-quired / Optional	Type	Description
uuid	Required	String	Resource manager UUID

Outputs

Name	Type	Description
auth_type	String	Authentication type. This is always **basic**.
hostname	String	Resource manager host name
password	String	Password to use for basic authentication
port	Integer	Resource manager port
type	String	Type of resource manager. This can be one of the following values. • **Schneider EcoStruxure IT** • **vRealize Operations Manager** • **XClarity Administrator**
username	String	User name to use for basic authentication
uuid	String	Resource manager UUID

The following example is returned if the request is successful.

```
{
    "auth_type": "basic",
    "hostname": "109.0.2.10,"
    "port": 443,
    "password": "password",
    "type": "XClarity Administrator",
    "username": "userid",
    "uuid": "8D735FCEFBCD49118C68169312166C68"
}
```

Connecting an XClarity Administrator resource manager

Use this definition to connect (add) a Lenovo XClarity Administrator resource manager to Lenovo XClarity Orchestrator.

Usage

```
resource "lxco_manager_resource" "string" {
    auth_type = "string"
    enabledriveanalytics = Boolean
    hostname = "string"
    password = "string"
    port = integer
    type = "string"
    username = "string"
}
```

Example

The following example connects an XClarity Administrator resource manager.

```
terraform {
   required_providers {
      lxco = {
         version = "0.1"
         source = "lenovo.com/xclarity/lxco"
      }
   }
}

provider "lxco" {
   host = "192.0.2.0"
   # username = ""
   # password = ""
}

# Connect an XClarity Administrator resource manager
resource "lxco_manager_resource" "LXCA_1" {
   auth_type = "basic"
   enabledriveanalytics = true
   hostname = "109.0.2.10"
   password = "********"
   port = 443
   type = "XClarity Administrator"
   username = "userid"
}
```

Inputs

Name	Re-quired / Optional	Type	Description
auth_type	Required	String	Authentication type. This is always **basic**.
enabledriveanalytics	Required	String	Indicates whether to enable collecting drive-analytics data daily for ThinkSystem and ThinkAgile devices. Drive analytics data is used for and is used for predictive analytics. This can be one of the following values. • **true**. Collect drive-analytics data daily. • **false**. Do not collect drive-analytics data.
hostname	Required	String	Resource manager host name
password	Required	String	Password to use for basic authentication
port	Required	Integer	Resource manager port
type	Required	String	Type of resource manager. This value is always **XClarity Administrator**.
username	Required	String	User name to use for basic authentication
uuid	Required	String	Resource manager UUID

Outputs

None

Connecting a Schneider Electric EcoStruxure IT Expert resource manager

Use this definition to connect (add) a Schneider Electric EcoStruxure IT Expert resource manager to Lenovo XClarity Orchestrator.

Usage

```
resource "lxco_manager_resource" "string" {
   auth_type = string
   name = string
   token = string
   type = string
   url = string
}
```

Example

The following example connects an EcoStruxure IT Expert resource manager.

```
terraform {
   required_providers {
      lxco = {
         version = "0.1"
         source = "lenovo.com/xclarity/lxco"
      }
   }
}

provider "lxco" {
   host = "192.0.2.0"
   # username = ""
   # password = ""
}

# Connect an EcoStruxure IT Expert resource manager
resource "lxco_manager_resource" "ECO_1" {
   auth_type = "token"
   name = "RM_1"
   token = "AK1/2sixskmmc06wj/1i6v3epcz5c25rc29jv1t00hce1pjahyobux63"
   type = "Schneider EcoStruxure IT"
   url = "https://api.ecostruxureit.com/rest/v1/organizations"
}
```

Inputs

Name	Re-quired / Optional	Type	Description
auth_type	Required	String	Authentication type. This is always **token**.
name	Required	String	Resource manager name
token	Required	String	Token value
type	Required	String	Type of resource manager. This value is always **Schneider EcoStruxure IT**.
url	Required	String	Resource manager URL

Outputs

None

Connecting a VMware vRealize Operations Manager resource manager

Use this definition to connect (add) a VMware vRealize Operations Manager resource manager to Lenovo XClarity Orchestrator.

Usage

```
resource "lxco_manager_resource" "string" {
   authS_source = "string"
   hostname = "string"
   password = "string"
   port = integer
   type = "string"
   username = "string"
}
```

Example

The following example connects a vRealize Operations Manager resource manager.

```
terraform {
   required_providers {
      lxco = {
         version = "0.1"
         source = "lenovo.com/xclarity/lxco"
      }
   }
}

provider "lxco" {
   host = "192.0.2.0"
   # username = ""
   # password = ""
}

# Connect a VMware vRealize Operations Manager resource manager
resource "lxco_manager_resource" "VROPS_1" {
   auth_source = "Local Users"
   hostname = "192.0.2.10"
   password = "********"
   port = 443
   type = "vRealize Operations Manager"
   username = "userid"
}
```

Inputs

Name	Re-quired / Optional	Type	Description
auth_source	Optional	String	Name of the authentication source for users and groups
hostname	Required	String	Resource manager host name
password	Required	String	Password to use for basic authentication
port	Required	Integer	Resource manager port

Name	Re-quired / Optional	Type	Description
type	Required	String	Type of resource manager. This value is always **vRealize Operations Manager**.
username	Required	String	User name to use for basic authentication

Outputs

None

Disconnecting a resource manager

Use the Terraform `destroy` command to disconnect (remove) a resource manager from Lenovo XClarity Orchestrator.

Usage

```
resource "lxco_manager" "destroy" {
   uuid = string
}
```

Example

The following example removes an XClarity Administrator resource manager.

```
terraform {
   required_providers {
      lxco = {
         version = "0.1"
         source = "lenovo.com/xclarity/lxco"
      }
   }
}

provider "lxco" {
   host = "192.0.2.0"
   # username = ""
   # password = ""
}

resource "lxco_manager" "destroy" {
   uuid = "8D735FCEFBCD49118C68169312166C68"
}
```

Inputs

Name	Re-quired / Optional	Type	Description
uuid	Required	String	Resource manager UUID

Outputs

None

Chapter 5. Managing servers

You can use Terraform configurations to retrieve information about servers, and to change the power state of a server.

Retrieving a list of all servers

Uses this definition to retrieve a list of all servers that are managed by Lenovo XClarity Orchestrator.

Usage

```
data "lxco_server" "all" {}

output "all_servers" {
   value = data.lxco_server.all
}
```

Example

The following example retrieves a list of all servers.

```
terraform {
   required_providers {
      lxco = {
         version = "0.1"
        source = "lenovo.com/xclarity/lxco"
      }
   }
}

provider "lxco" {
   host = "192.0.2.0"
   # username = ""
   # password = ""
}

# Fetch all server details
data "lxco_server" "all" {}

# Return details for all servers
output "all_servers" {
   value = data.lxco_server.all
}
```

Inputs

None

Outputs

Name		Type	Description
servers		Array of objects	Information about each managed server
	domain_name	String	Domain name
	group_names	Array of strings	List of names of the groups to which the device belongs

Name	Type	Description
health_status	String	Health state of the device (translated). This can be one of the following values. • **Normal** • **Warning** • **Critical** • **Unknown**
hostname	String	Hostname
id	String	Device ID
ipv4_addresses	Array of strings	List of IPv4 addresses
ipv6_addresses	Array of strings	List of IPv6 addresses
location	String	Location description
location_lowest_rack_unit	String	Lowest rack unit
location_rack	String	Rack name
location_room	String	Room name
machine_type	String	Device machine type
manager_domain_name	String	Fully qualified domain name
manager_hostname	String	Host name
manager_ipv4_addresses	Array of strings	IPv4 addresses
manager_ipv6_addresses	Array of strings	IPv6 addresses
model	String	Device model
power_status	String	Power status (translated). This can be one of the following values. • **Off** • **On** • **Standby** • **Unknown**
product_name	String	Device product name
serial_number	String	Device serial number
user_defined_name	String	User-defined name for the device
uuid	String	Device UUID

The following example is returned if the request is successful.

```
{
   "servers": [{
      "domain_name": "labs.lenovo.com",
      "group_names": [],
      "health_status": "Warning",
      "hostname": "IMM2-3440b5e913f8",
      "id": "FAAC1AC51EE411E3A8503440B5EAC7F0-23C87F0A2CB6491097489193447A655C",
      "ipv4_addresses": ["10.243.10.193","169.254.95.118"],
      "ipv6_addresses": ["2000:1013:0:0:0:0:217:105","fd55:faaf:e1ab:2021:3640:b5ff:fee9:13f8",
                         "fe80:0:0:0:3640:b5ff:fee9:13f8"],
      "location": "Morrisville",
      "location_lowest_rack_unit": 0,
      "location_rack": "",
      "location_room": "",
      "machine_type": "7916",
      "manager_domain_name": "labs.lenovo.com",
      "manager_hostname": "xhmc194",
      "manager_ipv4_addresses": ["10.243.2.107"],
      "manager_ipv6_addresses": ["fd55:faaf:e1ab:2021:5054:ff:fec4:df97","fe80:0:0:0:5054:ff:fec4:df97"],
      "model": "AC1",
      "power_status": "On",
      "product_name": "IBM Flex System x222 Lower Compute Node with embedded 10Gb Virtual Fabric",
      "serial_number": "SLOT002",
      "user_defined_name": "*node02_1",
      "uuid": "FAAC1AC51EE411E3A8503440B5EAC7F0"
   },
   {
      "domain_name": "labs.lenovo.com",
      "group_names": [],
      "health_status": "Warning",
      "hostname": "IMM2-3440b5ee128c",
      "id": "CB62A8381EEF11E387D53440B5EFC518-23C87F0A2CB6491097489193447A655C",
      "ipv4_addresses": ["10.243.11.11","169.254.95.118"],
      "ipv6_addresses": ["2000:1013:0:0:0:0:217:105","fd55:faaf:e1ab:2021:3640:b5ff:feee:128c",
                         "fe80:0:0:0:3640:b5ff:feee:128c"],
      "location": "Morrisville",
      "location_lowest_rack_unit": 0,
      "location_rack": "",
      "location_room": "",
      "machine_type": "7916",
      "manager_domain_name": "labs.lenovo.com",
      "manager_hostname": "xhmc194",
      "manager_ipv4_addresses": ["10.243.2.107"],
      "manager_ipv6_addresses": ["fd55:faaf:e1ab:2021:5054:ff:fec4:df97","fe80:0:0:0:5054:ff:fec4:df97"],
      "model": "99X",
      "power_status": "On",
      "product_name": "IBM Flex System x222 Upper Compute Node with embedded 10Gb Virtual Fabric",
      "serial_number": "SLOT002",
      "user_defined_name": "*node02_2",
      "uuid": "CB62A8381EEF11E387D53440B5EFC518"
   }]
}
```

Retrieving information about a specific server

Uses this definition to retrieve information about a specific server that is managed by Lenovo XClarity Orchestrator.

Usage

```
resource "lxco_server" "server" {
   resource_id = string
}
output "server" {
   value = data.lxco_server.server
}
```

Example

The following example retrieves information about a specific server.

```
terraform {
   required_providers {
      lxco = {
         version = "0.1"
        source = "lenovo.com/xclarity/lxco"
      }
   }
}

provider "lxco" {
   host = "192.0.2.0"
   # username = ""
   # password = ""
}

# Fetch details for a specific server
resource "lxco_server" "server" {
   resource_uuid = "00632D78DE644E23B712E200FE449787-7AF5D198CECF431AAEC674C7CA5A29B5"
}

# Return details for the server
output "server" {
   value = data.lxco_server.server
}
```

Inputs

Name	Re-quired / Optional	Type	Description
resource_uuid	Required	String	Server ID The device ID includes the UUID of the device and the UUID of the resource manager that manages the device, separated by a dash (*deviceUUID–managerUUID*).

Outputs

Name		Type	Description
servers		Array of objects	Information about the managed server
	domain_name	String	Domain name
	group_names	Array of strings	List of names of the groups to which the device belongs
	health_status	String	Health state of the device (translated). This can be one of the following values. • **Normal** • **Warning** • **Critical** • **Unknown**
	hostname	String	Hostname
	id	String	Device ID
	ipv4_addresses	Array of strings	List of IPv4 addresses
	ipv6_addresses	Array of strings	List of IPv6 addresses

Name	Type	Description
location	String	Location description
location_lowest_rack_unit	String	Lowest rack unit
location_rack	String	Rack name
location_room	String	Room name
machine_type	String	Device machine type
manager_domain_name	String	Fully qualified domain name
manager_hostname	String	Host name
manager_ipv4_addresses	Array of strings	IPv4 addresses
manager_ipv6_addresses	Array of strings	IPv6 addresses
model	String	Device model
power_status	String	Power status (translated). This can be one of the following values. • **Off** • **On** • **Standby** • **Unknown**
product_name	String	Device product name
serial_number	String	Device serial number
user_defined_name	String	User-defined name for the device
uuid	String	Device UUID

The following example is returned if the request is successful.

```
{
   "servers":[{
      "domain_name": "labs.lenovo.com",
      "group_names": [],
      "health_status": "Warning",
      "hostname": "IMM2-3440b5e913f8",
      "id": "FAAC1AC51EE411E3A8503440B5EAC7F0-23C87F0A2CB6491097489193447A655C",
      "ipv4_addresses": ["10.243.10.193","169.254.95.118"],
      "ipv6_addresses": ["2000:1013:0:0:0:0:217:105","fd55:faaf:e1ab:2021:3640:b5ff:fee9:13f8",
                         "fe80:0:0:0:3640:b5ff:fee9:13f8"],
      "location": "Morrisville",
      "location_lowest_rack_unit": 0,
      "location_rack": "",
      "location_room": "",
      "machine_type": "7916",
      "manager_domain_name": "labs.lenovo.com",
      "manager_hostname": "xhmc194",
      "manager_ipv4_addresses": ["10.243.2.107"],
      "manager_ipv6_addresses": ["fd55:faaf:e1ab:2021:5054:ff:fec4:df97","fe80:0:0:0:5054:ff:fec4:df97"],
      "model": "AC1",
      "power_status": "On",
      "product_name": "IBM Flex System x222 Lower Compute Node with embedded 10Gb Virtual Fabric",
      "serial_number": "SLOT002",
      "user_defined_name": "*node02_1",
      "uuid": "FAAC1AC51EE411E3A8503440B5EAC7F0"
   }]
}
```

Changing the power state of servers

Uses this definition to perform a power action on servers that are managed by Lenovo XClarity Orchestrator and then return the power status of that server.

Usage

```
resource "lxco_server" "power_action" {
   group_ids = array of strings
   power_action = string
   resource_ids = array of strings
}

output "power_action" {
   value = lxco_server.power_action
}
```

Example

The following example restarts two server and powers on a group of servers.

```
terraform {
   required_providers {
      lxco = {
         version = "0.1"
        source = "lenovo.com/xclarity/lxco"
      }
   }
}

provider "lxco" {
   host = "192.0.2.0"
   # username = ""
   # password = ""
}

# Power on specific servers, and return the results
resource "lxco_server" "power_action" {
   group_ids = [],
   resource_ids = [
      "80CE6AB8FF7D11E685CB819F6B26BCF8-AC2E339A942446F4A246BB55B56FB18F"
      "00632D78DE644E23B712E200FE449787-97E61EA441F3491B9CC971E68D2D8BCD"
   ],
   power_action = "PowerOn"
}
output "power_action" {
   value = lxco_server.power_action
}

# Restart a group of servers, and return the results
resource "lxco_server" "power_action" {
   group_ids = ["G_36898672B78D4A93B2829123E7728925"]
   resource_ids = [],
   power_action = "Restart"
}
output "power_action" {
   value = lxco_server.power_action
}
```

Inputs

Name	Re-quired / Optional	Type	Description
group_ids	Required	Array of strings	List of device group IDs Specify an empty array if the **resource_ids** attribute is specified.
power_action	Required	String	Power action. This can be one of the following values. • **PowerOn**. Powers on the resource. • **PowerOff**. Powers off the resource immediately. • **PowerOffSoft**. Shuts down the operating system and powers off the resource. • **Restart**. Restarts the resource immediately. • **RestartSoft**. Shuts down the operating system and restarts the resource. • **RestartBMC**. Restarts the baseboard management controller. • **BootToF1**. Restarts the resource to BIOS/UEFI (F1) Setup. This is supported for non-ThinkServer servers that are supported without limitations.
resource_ids	Required	Array of strings	List of resources IDs Specify an empty array if the **group_ids** attribute is specified.

Outputs

Name	Type	Description
id	String	Job ID
power_action	String	Power action that was performed
resource_ids	Array of strings	List of IDs of resources on which the power action was performed
status	Integer	Message about the status of the action

The following example is returned if the request is successful.

```
{
   "id":"0349DC28D0C411E7B5A47ED30AE32DCF",
   "power_action": "PowerOn",
   "resource_ids": [
      "80CE6AB8FF7D11E685CB819F6B26BCF8-AC2E339A942446F4A246BB55B56FB18F"
      "00632D78DE644E23B712E200FE449787-97E61EA441F3491B9CC971E68D2D8BCD"
   ],
   "status": "The power action job launched successfully. The power action job was launched in the
              orchestrator server and will be executed asynchronously. Job ID: 125"
}
```

Chapter 6. Provisioning updates to managed resources

You can use Terraform configurations to maintain current software levels on Lenovo XClarity Orchestrator resource managers and managed servers. You can use update-compliance policies to identify which resources need to be updated based on custom criteria.

Creating an update-compliance policy

Use this definition to create an update-compliance policy.

A job is created to complete this request.

Usage

```
resource "lxco_firmware" "createPolicy" {
   policy_action = "createPolicy"
   name = string
   description = string
   compliance_rule = string
   rules {
      platformidentifier = string
      criteria {
         targetcomponentid = string
         targetupdatepackageid = string
      }
      criteria {
      targetcomponentid = string
      targetupdatepackageid = string
      }
      criteria {
         targetcomponentid = string
         targetupdatepackageid = string
      }
   }
}

output "createPolicy" {
   value = lxco_firmware.createPolicy
}
```

Example

The following example creates a policy.

```
terraform {
   required_providers {
      lxco = {
         version = "0.1"
        source = "lenovo.com/xclarity/lxco"
      }
   }
}

provider "lxco" {
   host = "192.0.2.0"
   # username = ""
   # password = ""
}
```

```
# Create a firmware-compliance policy, and return the results
resource "lxco_firmware" "createPolicy" {
   policy_action = "createPolicy"
   name = "TestPolicy05"
   description = "TestPolicy05"
   compliance_rule = "FlagIfNotExactMatch"
   rules {
      platformidentifier = "lxca"
      criteria {
         targetcomponentid = "LXPM-7X04"
         targetupdatepackageid = "lnvgy_fw_lxpm_pdl132e-2.03_anyos_noarch"
      }
      criteria {
         targetcomponentid = "UEFI-7X04"
         targetupdatepackageid = "lnvgy_fw_uefi_tee168j-2.91_anyos_32-64"
      }
      criteria {
         targetcomponentid     = "XCC-7X04"
         targetupdatepackageid = "lnvgy_fw_xcc_cdi376s-6.60_anyos_noarch"
      }
   }
}

output "createPolicy" {
   value = lxco_firmware.createPolicy
}
```

Inputs

Name		Re-quired / Optional	Type	Description
name		Required	String	Policy name
description		Required	String	Policy description
compliance_rule		Required	String	Indicates when to flag a resource as non-compliant. This can be one of the following values. • **DoNotFlag**. Devices that are out of compliance are not flagged • **FlagIfNotExactMatch**. If the software or firmware level that is installed on a resource is not an exact match with the update-compliance policy, the resource is flagged as non-compliant. For example, if you replace a network adapter in a server, and the firmware on that network adapter is different than the level identified in the update-compliance policy, then the server is flagged as non-compliant.
rules		Required	String	Information about each rule for this policy
	platformidentifier	Required	String	ID of the platform (resource type) that is associated with the target component
	criteria	Required	String	Information about the target component and update for this policy You can specify one or more criteria objects, one for each target component. Tip: If the platform does not have components, specify the platform ID.

Name			Re-quired / Optional	Type	Description
		targetcomponentid	Required	String	Target component ID
		targetupdatepackageid	Required	String	Target update package ID

Outputs

Name	Type	Description
id	String	Job ID
status	String	Message about the status of the action

The following example is returned if the request is successful.

```
{
   "id": "125",
   "status": "The job was created successfully. The job was launched in the
             orchestrator server and will be run asynchronously. Job ID: 125"
}
```

Modifying an update-compliance policy

Uses this definition to modify an update-compliance policy.

A job is created to complete this request.

Usage

```
resource "lxco_firmware" "updatePolicy" {
   policy_action = "updatePolicy"
   policy_id = "12980764"
   name = string
   description = string
   compliance_rule = string
   rules {
      platformidentifier = string
      criteria {
         targetcomponentid = string
         targetupdatepackageid = string
      }
      criteria {
      targetcomponentid = string
      targetupdatepackageid = string
      }
      criteria {
         targetcomponentid = string
         targetupdatepackageid = string
      }
   }
}

output "updatePolicy" {
   value = lxco_firmware.updatePolicy
}
```

Example

The following example creates a policy.

```
terraform {
   required_providers {
      lxco = {
         version = "0.1"
        source = "lenovo.com/xclarity/lxco"
      }
   }
}

provider "lxco" {
   host = "192.0.2.0"
   # username = ""
   # password = ""
}

# Create a firmware-compliance policy, and return the results
resource "lxco_firmware" "updatePolicy" {
   policy_action = "updatePolicy"
   policy_id = "12980764"
   name = "TestPolicy05"
   description = "TestPolicy05"
   compliance_rule = "FlagIfNotExactMatch"
   rules {
      platformidentifier = "lxca"
      criteria {
         targetcomponentid = "LXPM-7X04"
         targetupdatepackageid = "lnvgy_fw_lxpm_pdl132e-2.03_anyos_noarch"
      }
      criteria {
         targetcomponentid = "UEFI-7X04"
         targetupdatepackageid = "lnvgy_fw_uefi_tee168j-2.91_anyos_32-64"
      }
      criteria {
         targetcomponentid     = "XCC-7X04"
         targetupdatepackageid = "lnvgy_fw_xcc_cdi376s-6.60_anyos_noarch"
      }
   }
}

output "updatePolicy" {
   value = lxco_firmware.updatePolicy
}
```

Inputs

Name	Re-quired / Optional	Type	Description
policy_id	Required	String	Policy ID
name	Required	String	Policy name
description	Required	String	Policy description

Name	Re-quired / Optional	Type	Description
compliance_rule	Required	String	Indicates when to flag a resource as non-compliant. This can be one of the following values. • **DoNotFlag**. Devices that are out of compliance are not flagged • **FlagIfNotExactMatch**. If the software or firmware level that is installed on a resource is not an exact match with the update-compliance policy, the resource is flagged as non-compliant. For example, if you replace a network adapter in a server, and the firmware on that network adapter is different than the level identified in the update-compliance policy, then the server is flagged as non-compliant.
rules	Required	String	Information about each rule for this policy
platformidentifier	Required	String	ID of the platform (resource type) that is associated with the target component
criteria	Required	String	Information about the target component and update for this policy You can specify one or more criteria objects, one for each target component. Tip: If the platform does not have components, specify the platform ID.
targetcomponentid	Required	String	Target component ID
targetupdatepackageid	Required	String	Target update package ID

Outputs

Name	Type	Description
id	String	Job ID
status	String	Message about the status of the action

The following example is returned if the request is successful.

```
{
   "id": "125",
   "status": "The job was created successfully. The job was launched in the
             orchestrator server and will be run asynchronously. Job ID: 125"
}
```

Assigning an update-compliance policy to a group of resources

Uses this definition to assign an update-compliance policy to a group of resources.

A job is created to complete this request.

Usage

```
resource "lxco_firmware" "assignPolicy" {
   policy_action = "assignPolicy"
   group_ids = array of strings
   resource_ids = array of strings
   overwrite = BBoolean
   policy_id = string
}

output "assignPolicy" {
   value = lxco_firmware.assignPolicy
}
```

Example

The following example assigns a policy to a group of serves and a specific server.

```
terraform {
   required_providers {
      lxco = {
         version = "0.1"
        source = "lenovo.com/xclarity/lxco"
      }
   }
}

provider "lxco" {
   host = "192.0.2.0"
   # username = ""
   # password = ""
}

# Assign a firmware-compliance policy, and return the results
resource "lxco_firmware" "assignPolicy" {
   policy_action = "assignPolicy"
   group_ids = ["G_E261C2F34895442482F7D638BA40F964"]
   resource_ids = ["80CE6AB8FF7D11E685CB819F6B26BCF8-C3B280177A194899B5C122118EDFB944"]
   overwrite = true
   policy_id = "1631709885143"
}

output "assignPolicy" {
   value = lxco_firmware.assignPolicy
}
```

Inputs

Name	Re-quired / Optional	Type	Description
policy_id	Required	String	Policy ID
group_ids	Required	Array of strings	List of IDs of resource and device groups The default is an empty array.
resource_ids	Required	Array of strings	List of IDs of resource managers and managed devices The default is an empty array.
overwrite	Required	Boolean	Indicates whether the policy for the platform is changed if another policy is assigned to that platform. This can be one of the following values. • **true**. If another policy is assigned to the platform, the policy is changed. • **false**. (default) If another policy is assigned to the platform, the policy is not changed.

Outputs

Name	Type	Description
id	String	Job ID
status	String	Message about the status of the action

The following example is returned if the request is successful.

```
{
   "id": "125",
   "status": "The job was created successfully. The job was launched in the
             orchestrator server and will be run asynchronously. Job ID: 125"
}
```

Applying an update-compliance policy to one or more resources

Uses this definition to apply an update-compliance policy to one or more resources.

A job is created to complete this request.

Usage

```
resource "lxco_firmware" "applyAndactivate" {
   policy_action = "applyAndactivate"
   activation_rule = string
   force_update = Boolean
   group_ids = array of strings
   install_prerequisite_firmware = Boolean
   policy_id = string
   resource_ids = array of strings
   update_rule = string
}

output "applyAndactivate" {
   value = lxco_firmware.applyAndactivate
}
```

Example

The following example assigns a policy to a group of serves and a specific server.

```
terraform {
   required_providers {
      lxco = {
         version = "0.1"
        source = "lenovo.com/xclarity/lxco"
      }
   }
}

provider "lxco" {
   host = "192.0.2.0"
   # username = ""
   # password = ""
}

# Apply a firmware-compliance policy, and return the results
resource "lxco_firmware" "applyAndactivate" {
   policy_action = "applyAndactivate"
   activation_rule = "ImmediateActivation"
   force_update = true
   group_ids = ["G_162B69BD175947CC9AAD0E0C7CB6045C"]
   install_prerequisite_firmware = false
   policy_id = "1633679392153"
   resource_ids = []
   update_rule = "ContinueOnError"
}

output "applyAndactivate" {
   value = lxco_firmware.applyAndactivate
}
```

Inputs

Name	Re-quired / Optional	Type	Description
activation_rule	Required	String	(Managed devices only) Indicates when to activate the update. This can be one of the following values. • **ImmediateActivation**. (default) During the update process, the resource might be restarted automatically several times until the entire process is complete. Ensure that you quiesce all applications on the resource before you proceed. • **DelayedActivation**. (Servers and RackSwitch devices only) Some but not all update operations are performed. Resources must be restarted manually to continue the update process. Additional restarts are then performed until the update operation completes. • **PrioritizedActivation**. (Servers and RackSwitch devices only) Baseboard Management Controller is applied and activated immediately while other firmware is performed in delayed activation mode. For resource managers, the update is activated immediately. The resource might be restarted automatically several times until the entire process is complete. Ensure that you quiesce all applications on the resource before you proceed.
force_update	Required	Boolean	(Managed devices only) Indicates whether to apply the update to selected components even if the current software or firmware level is up to date or to apply an update that is earlier than the one that is currently installed. **Important:** You cannot apply earlier levels of firmware to device options, adapters, and disk drives. This can be one of the following values. • **true**. Applies the update to the selected resources even if the software or firmware is compliant. • **false**. (default) Skips the update on the selected resources if the software or firmware is already compliant. For resource managers, you cannot apply an update of the same or earlier software level as the one that is currently installed on a resource manager.

Name	Re- quired / Optional	Type	Description
group_ids	Required	Array of strings	List of IDs of resource group to which to apply the updates The update is applied to each resource in the specified groups only if the resource has an assigned compliance policy and is out of compliance with that policy. Set this attribute to null if you do not want to specify a group. If **resource_ids** and **group_ids** are set to an empty array and **policy_id** is set to null, all managed resources that are not compliant with their assigned policy are updated by default.
install_prerequisite_firmware	Required	Boolean	(Managed devices only) Indicates whether to install prerequisite updates. Prerequisite updates are installed and activated before the remaining updates are installed and activated. Multiple reboots might be required to install all updates. This can be one of the following values. • **true**. (default) Installs all prerequisite updates, if needed. • **false**. Do not install prerequisite updates. For resource managers, prerequisite updates are not applied automatically to resource managers.
policy_id	Optional	String	ID of the update-compliance policy The update is applied to each resource that is assigned the policy only if the resource is out of compliance with that policy. If you specify a **policy_id**, you must set **resource_ids** to an empty array. You can specify either **policy_id** or resources using **resource_ids**, but not both. If **resource_ids** and **group_ids** are set to an empty array and **policy_id** is set to null, all managed resources that are not compliant with their assigned policy are updated by default.

Name	Re-quired / Optional	Type	Description
resource_ids	Required	Array of strings	List of IDs of resources to which to apply the updates The update is applied to each specified resource only if the resource has an assigned compliance policy and is out of compliance with that policy. Set this attribute to an empty array if you do not want to specify a resource. If you specify resources using **resource_ids**, you cannot specify **policy_id**. If **resource_ids** and **group_ids** are set to an empty array and **policy_id** is set to null, all managed resources that are not compliant with their assigned policy are updated by default.
update_rule	Required	String	(Managed devices only) Indicates how to handle errors during the update process. This can be one of the following values. • **ContinueOnError**. If an error occurs when updating one of the components in a resource (such as an adapter or management-controller firmware), the update process does not apply the update for that specific component. However, the orchestrator server continues to update other components for the resource and continues with all other updates in the current update job. • **AbortOnError**. If an error occurs when updating one of the components in a resource (such as an adapter or management-controller firmware), the updates process stops the remaining updates for that specific resource. The current firmware that is installed on that resource remains in effect. However, current update job includes additional resources, the orchestrator server continues to update the remaining resources. For resource managers, the update always continues on error.

Outputs

Name	Type	Description
id	String	Job ID
status	String	Message about the status of the action

The following example is returned if the request is successful.

```
{
   "id":"125",
   "status": "The job was created successfully. The job was launched in the
             orchestrator server and will be run asynchronously. Job ID: 125"
}
```