NSX-T Terraform Provider 3.3.2 Release Notes

NSX-T Terraform Provider 3.3.2 | October 2023

What's in the Release Notes

What is New

This release of NSX-T Terraform Provider 3.3.2 offers a number of Improvements and Bugfixes which will be described below. It introduces support for multi-tenancy with ability to switch context between Projects and also adds some experimental features.

We also remind that aligned with the NSX platform policy the resources/data-sources targeting Imperative API (also called MP APIs) are deprecated

The certified provider can be downloaded from Hashicorp directly via Terraform or found at: https://github.com/vmware/terraform-provider-nsxt

For all the features available in Federation this is described in detail in the provider documentation under **Guides** :
https://registry.terraform.io/providers/vmware/nsxt/latest/docs/guides/federation

For all the features available on VMConAWS this is described in detail in the provider documentation under
Guides:https://registry.terraform.io/providers/vmware/nsxt/latest/docs/guides/vmc

All these resources and data sources are fully documented on the NSX-T Terraform Provider page: https://www.terraform.io/docs/providers/nsxt/index.html
For more details on the NSX-Policy API usage, please look at NSX-T documentation.

What is There

As in Terraform 3.3.2 the following are supported:
Additions are marked in green (like **nsxt_policy_dhcp_server**). The number of new resources / datasources is limited since we have enhanced existing ones to support Federation.

Data Sources

Federation

- nsxt_policy_site

Policy API data sources

- nsxt_policy_certificate
- nsxt_policy_edge_cluster
- nsxt_policy_edge_node
- nsxt_policy_ip_discovery_profile
- nsxt_policy_ipv6_dad_profile
- nsxt_policy_ipv6_ndra_profile
- nsxt_policy_lb_app_profile
- nsxt_policy_lb_client_ssl_profile
- nsxt_policy_lb_monitor
- nsxt_policy_lb_persistence_profile
- nsxt_policy_lb_server_ssl_profile
- nsxt_policy_mac_discovery_profile
- nsxt_policy_qos_profile
- nsxt_policy_realization_info
- nsxt_policy_segment_security_profile
- nsxt_policy_segment_realization
- nsxt_policy_service
- nsxt_policy_spoofguard_profile
- nsxt_policy_tier0_gateway
- nsxt_policy_tier1_gateway
- nsxt_policy_transport_zone
- nsxt_policy_vm
- nsxt_policy_vni_pool
- nsxt_policy_security_policy
- nsxt_policy_gateway_policy
- nsxt_policy_group
- nsxt_policy_context_profile
- nsxt_management_cluster
- nsx_policy_bfd_profile
- nsxt_policy_intrusion_service_profile (Local Manager only)
- nsxt_policy_lb_service
- nsxt_policy_dhcp_server
- nsxt_policy_ip_block
- nsxt_policy_gateway_locale_service
- nsxt_policy_bridge_profile

- nsxt_policy_l2_vpn_service
- nsxt_policy_ipsec_vpn_service
- nsxt_policy_ipsec_vpn_local_endpoint
- nsxt_policy_vms.
- nsxt_policy_lb_service
- nsxt_policy_segment

Imperative API data sources (Deprecated)

- nsxt_edge_cluster
- nsxt_logical_tier0_router
- nsxt_ns_service
- nsxt_switching_profile
- nsxt_transport_zone
- nsxt_certificate
- nsxt_mac_pool
- nsxt_ns_group
- nsxt_ns_service
- nsxt_ns_groups
- nsxt_ns_services

Resources

Policy API Resources – Segments

- nsxt_policy_segment
- nsxt_policy_vlan_segment
- nsxt_policy_fixed_segment (VMC only).
- nsxt_policy_qos_profile
- nsxt_policy_mac_discovery_profile
- nsxt_policy_ip_discovery_profile
- nsxt_policy_segment_security_profile
- nsxt_policy_spoof_guard_profile

Policy API Resources – Gateways & Routing

- nsxt_policy_tier0_gateway
- nsxt_policy_tier0_gateway_interface
- nsxt_policy_tier1_gateway
- nsxt_policy_tier1_gateway_interface
- nsxt_policy_static_route
- nsxt_policy_bgp_neighbor
- nsxt_policy_bgp_config
- nsxt_policy_ospf_config
- nsxt_policy_ospf_area
- nsxt_policy_nat_rule
- nsxt_policy_gateway_prefix_list
- nsxt_policy_tier0_gateway_ha_vip_config
- nsxt_policy_gateway_community_list
- nsxt_policy_gateway_route_map

- nsxt_policy_static_route_bfd_peer
- nsxt_policy_gateway_redistribution_config
- nsxt_policy_predefined_gateway_policy
- nsxt_policy_gateway_qos_profile

Policy API Resources – EVPN

- nsxt_policy_evpn_tenant
- nsxt_policy_evpn_config
- nsxt_policy_evpn_tunnel_endpoint

Policy API Resources – Firewall (Centralized and Distributed)

- nsxt_policy_gateway_policy
- nsxt_policy_security_policy
- nsxt_policy_service
- nsxt_policy_context_profile
- nsxt_policy_predefined_security_policy
- nsxt_policy_context_profile_custom_attribute

Policy API Resources – IDS/IPS

- nsxt_policy_intrusion_service_profile
- nsxt_policy_intrusion_service_policy

Policy API Resources – Grouping & Tagging

- nsxt_policy_group
- nsxt_policy_vm_tags

Policy API Resources – Load Balancer

- nsxt_policy_lb_pool
- nsxt_policy_lb_service
- nsxt_policy_lb_virtual_server

Policy API Resources – DNS Forwarder

- nsxt_policy_dns_forwarder_zone
- nsxt_policy_gateway_dns_forwarder

Policy API Resources – DHCP

- nsxt_policy_ip_address_allocation
- nsxt_policy_ip_block

Policy API Resources – IP allocation

- nsxt_policy_ip_address_allocation

- nsxt_policy_ip_block
- nsxt_policy_ip_pool
- nsxt_policy_ip_pool_block_subnet
- nsxt_policy_ip_pool_static_subnet
- nsxt_policy_ip_pool_test

Policy API Resources – VPN

- nsxt_policy_ipsec_vpn_dpd_profile
- nsxt_policy_ipsec_vpn_ike_profile
- nsxt_policy_ipsec_vpn_local_endpoint
- nsxt_policy_ipsec_vpn_service
- nsxt_policy_ipsec_vpn_session
- nsxt_policy_ipsec_vpn_tunnel_profile
- nsxt_policy_l2_vpn_service
- nsxt_policy_l2_vpn_session

Experimental Features:

- data/nsxt_policy_gateway_prefix_list
- data/nsxt_policy_gateway_route_map
- data/nsxt_policy_project
- resource/nsxt_policy_vni_pool
- resource/nsxt_policy_project
- Multitenancy support in selected resources, controlled by `context` argument
- Fabric resources and data sources (detailed list coming with next feature release)

Imperative API logical port and switching profile (Deprecated)

- nsxt_logical_port
- nsxt_mac_management_switching_profile
- nsxt_ip_discovery_switching_profile
- nsxt_qos_switching_profile
- nsxt_spoofguard_switching_profil
- nsxt_switch_security_switching_profile

Imperative API logical Switch (L2) (Deprecated)

- nsxt_logical_switch
- nsxt_vlan_logical_switch

Imperative API logical Router (L3) (Deprecated)

- nsxt_logical_tier0_router
- nsxt_logical_tier1_router
- nsxt_logical_router_downlink_port
- nsxt_logical_router_link_port_on_tier0
- nsxt_logical_router_link_port_on_tier1
- nsxt_logical_router_centralized_service_port

- nsxt_nat_rule
- nsxt_static_route

Imperative API DHCP and DHCP relay (Deprecated)

- nsxt_logical_dhcp_port
- nsxt_logical_dhcp_server
- nsxt_dhcp_server_ip_pool
- nsxt_dhcp_server_profile
- nsxt_dhcp_relay_profile
- nsxt_dhcp_relay_service

Imperative API load Balancer (Deprecated)

- nsxt_lb_service
- nsxt_lb_pool
- nsxt_lb_tcp_virtual_server
- nsxt_lb_udp_virtual_server
- nsxt_lb_http_virtual_server
- nsxt_lb_fast_tcp_application_profile
- nsxt_lb_fast_udp_application_profile
- nsxt_lb_http_application_profile
- nsxt_lb_client_ssl_profile
- nsxt_lb_http_request_rewrite_rule
- nsxt_lb_http_response_rewrite_rule
- nsxt_lb_http_forwarding_rule
- nsxt_lb_cookie_persistence_profile
- nsxt_lb_server_ssl_profile
- nsxt_lb_source_ip_persistence_profile
- nsxt_lb_http_monitor
- nsxt_lb_https_monitor
- nsxt_lb_icmp_monitor
- nsxt_lb_passive_monitor
- nsxt_lb_tcp_monitor
- nsxt_lb_udp_monitor

Imperative API firewall and grouping objects (Deprecated)

- nsxt_firewall_section
- nsxt_ip_set
- nsxt_ns_group
- nsxt_ns_service_group
- nsxt_vm_tags
- nsxt_ip_block
- nsxt_ip_block_subnet

Imperative API NS Service Resources (Deprecated)

- nsxt_algorithm_type_ns_service
- nsxt_ether_type_ns_service

- nsxt_icmp_type_ns_service
- nsxt_igmp_type_ns_service
- nsxt_ip_protocol_ns_service
- nsxt_l4_port_set_ns_service
- nsxt_igmp_type_ns_service

Improvements

- Support on-demand connection init in the provider. This behavior is controlled with `on_demand_connection` flag and is useful is NSX manager is not available at the time of plan/apply (#948)
- `resource/nsxt_policy_tier1_gateway`: Support `type` argument. This argument helps with auto-configuring route advertisements and provides the user experience that is consistent with UI on VMC (#909)
- Improve debug logging by dumping NSX API requests and responses when `TF_LOG_PROVIDER_NSX_HTTP` env variable is set (#963)

Bug Fixes

- `resource/nsxt_policy_security_policy`, `resource/nsxt_policy_gateway_policy`, `resource/nsxt_policy_ids_policy`: Fix rule ordering issue by auto-assigning sequence_number. (#967)
- `resource/nsxt_policy_group`: Fix `group_type` assignment on VMC by using node/version API to determine underlying NSX version (#970)
- `resource/nsxt_nat_rule`: Ensure compatibility with NSX 4.1.0 and above by replacing removed 'nat_pass' property with 'firewall_match' (#950)

Deprecations

As a reminder the non-policy data sources and resources (Imperatibe API also called MP APIs) are deprecated. Please use corresponding policy resources instead.

System Requirements

The NSX-T Terraform Provider 3.3.2 supports fully NSX version 3.2.x and 4.0.x and Terraform 0.12 onward or above. The recommended vSphere provider to be used in conjunction with the NSX-T Terraform Provider is 1.3.3 or above.

The NSX-T Terraform Provider 3.3.2 is backward compatible with NSX-T 3.1.x and 3.0.x and will work with previous versions except otherwise indicated on the resource.

Installation

Automated Installation (Recommended)

Download and initialization of Terraform providers is with the “terraform init” command. This applies to the NSX-T provider as well. Once the provider block for the NSX-T provider is specified in your .tf file, “terraform init” will detect a need for the provider and download it to

your environment. You can list versions of providers installed in your environment by running "terraform version" command:

```
$ ./terraform version
Terraform v0.13.0
+ provider.nsxt v3.2.5
+ provider.vsphere v1.5.0
```

Manual Installation

NOTE: Unless you are developing or require a pre-release bugfix or feature, you will want to use the officially released version of the provider (see the section above).

NOTE: Note that if the provider is manually copied to your running folder (rather than fetched with the "terraform init" based on provider block), Terraform is not aware of the version of the provider you're running. It will appear as "unversioned":

```
$ ./terraform version
Terraform v0.13.0
+ provider.nsxt (unversioned)
+ provider.vsphere v1.5.0
```

Since Terraform has no indication of version, it cannot upgrade in a native way, based on the "version" attribute in provider block. In addition, this may cause difficulties in housekeeping and issue reporting.

Cloning the Project

First, you will want to clone the repository to `$GOPATH/src/github.com/vmware/terraform-provider-nsxt`:

```
mkdir -p $GOPATH/src/github.com/vmware
cd $GOPATH/src/github.com/vmware
git clone https://github.com/vmware/terraform-provider-nsxt.git
```

Building and Installing the Provider

After the clone has been completed, you can enter the provider directory and build the provider.

```
cd $GOPATH/src/github.com/terraform-providers/terraform-provider-nsxt
make
```

After the build is complete, if your terraform running folder does not match your GOPATH environment, you need to copy the terraform-provider-nsxt executable to your running folder and re-run terraform init to make terraform aware of your local provider executable.

After this, your project-local `.terraform/plugins/ARCH/lock.json` (where ARCH matches the architecture of your machine) file should contain a SHA256 sum that matches the local plugin. Run `shasum -a 256` on the binary to verify the values match.

Known Issues

Functional Issues

- In the Imperative API backend does not allow changing a Tiers-1 edge cluster with single update call.
  It requires to disable firewall first, then disassociate the cluster and associate the new one. For now edge_cluster_id was marked as ForceNew attribute, meaning if edge_cluster_id is changed, the router would be recreated on backend.

- The Policy API can take some time to realize an intent. This can cause an issue when an existing entry hasn't been fully deleted and the same nsx_id is reused.
  The solution is to let the platform fully delete the object and reapply the Terraform configuration.