Thunder Terraform Provider

Version 1.1.0

February, 2023

Information in this document is subject to change without notice.

PATENT PROTECTION

A10 Networks, Inc. products are protected by patents in the U.S. and elsewhere. The following website is provided to satisfy the virtual patent marking provisions of various jurisdictions including the virtual patent marking provisions of the America Invents Act. A10 Networks, Inc. products, including all Thunder Series products, are protected by one or more of U.S. patents and patents pending listed at: a10-virtual-patent-marking.

TRADEMARKS

A10 Networks, Inc. trademarks are listed at: a10-trademarks

CONFIDENTIALITY

This document contains confidential materials proprietary to A10 Networks, Inc. This document and information and ideas herein may not be disclosed, copied, reproduced or distributed to anyone outside A10 Networks, Inc. without prior written consent of A10 Networks, Inc.

DISCLAIMER

This document does not create any express or implied warranty about A10 Networks, Inc. or about its products or services, including but not limited to fitness for a particular use and non-infringement. A10 Networks, Inc. has made reasonable efforts to verify that the information contained herein is accurate, but A10 Networks, Inc. assumes no responsibility for its use. All information is provided "as-is." The product specifications and features described in this publication are based on the latest information available; however, specifications are subject to change without notice, and certain features may not be available upon initial product release. Contact A10 Networks, Inc. for current information regarding its products or services. A10 Networks, Inc. products and services are subject to A10 Networks, Inc. standard terms and conditions.

ENVIRONMENTAL CONSIDERATIONS

Some electronic components may possibly contain dangerous substances. For information on specific component types, please contact the manufacturer of that component. Always consult local authorities for regulations regarding proper disposal of electronic components in your area.

FURTHER INFORMATION

For additional information about A10 products, terms and conditions of delivery, and pricing, contact your nearest A10 Networks, Inc. location, which can be found by visiting www.a10networks.com.

Table of Contents

Overview

Terraform is a tool for deploying and managing Infrastructure as Code (IaC). The A10 Thunder Terraform Provider (TTP) is a plug-in for provisioning and configuring objects for Thunder devices.

It includes both low-level components like compute instances, storage, and networking, as well as high-level components like DNS entries and SaaS features. It supports core features for ADC. For detailed information, see GitHub. For more information on Terraform, see https://www.terraform.io/intro.

NOTE: If you are familiar with Terraform, you can skip this quick start guide and check the documentation at terraform.registry for details regarding TTP.

Compatibility and Versioning

All ACOS 5.2.1-P4, 5.2.1-P5, and 5.2.1-P6 devices are supported with Thunder Terraform Provider 1.1.0.

Supported aXAPI

The following aXAPI endpoints are supported in TTP 1.1.0.

Table 1 : Supported aXAPI

Display Name	ACOS-AXAPI URI
Access List Extended	/axapi/v3/access-list/extended
Access List Standard	/axapi/v3/access-list/standard
Banner	/axapi/v3/banner
Bgp	/axapi/v3/bgp
Class List	/axapi/v3/class-list
Configure Sync	/axapi/v3/configure/sync

Display Name	ACOS-AXAPI URI
Ethernet	/axapi/v3/interface/ethernet/
File Aflex	/axapi/v3/file/aflex
File Bw List	/axapi/v3/file/bw-list
File Ca Cert	/axapi/v3/file/ca-cert
File Class List	/axapi/v3/file/class-list
File Class List Convert	/axapi/v3/file/class-list
File Csr	/axapi/v3/file/csr
File Ssl Cert	/axapi/v3/file/ssl-cert
File Ssl Cert Key	/axapi/v3/file/ssl-cert-key
File Ssl Crl	/axapi/v3/file/ssl-crl
File Ssl Key	/axapi/v3/file/ssl-key
Fw Active Rule Set	/axapi/v3/fw/active-rule-set
Fw Alg Dns	/axapi/v3/fw/alg/dns
Fw Alg Ftp	/axapi/v3/fw/alg/ftp
Fw Alg Icmp	/axapi/v3/fw/alg/icmp
Fw Alg Pptp	/axapi/v3/fw/alg/pptp
Fw Alg Rtsp	/axapi/v3/fw/alg/rtsp
Fw Alg Sip	/axapi/v3/fw/alg/sip
Fw Alg Tftp	/axapi/v3/fw/alg/tftp
Fw App	/axapi/v3/fw/app
Fw Apply Changes	/axapi/v3/fw/apply-changes
Fw Clear Session Filter	/axapi/v3/fw/clear-session-filter
Fw Full Cone Session	/axapi/v3/fw/full-cone-session
Fw Global	/axapi/v3/fw/global
Fw Gtp	/axapi/v3/fw/gtp
Fw Gtp In Gtp Filtering	/axapi/v3/fw/gtp-in-gtp-filtering
Fw Gtp V0	/axapi/v3/fw/gtp-v0
Fw Helper Sessions	/axapi/v3/fw/helper-sessions

Display Name	ACOS-AXAPI URI
Fw Limit Entry	/axapi/v3/fw/limit-entry
Fw Local Log	/axapi/v3/fw/local-log
Fw Logging	/axapi/v3/fw/logging
Fw Radius Server	/axapi/v3/fw/radius/server
Fw Usage	/axapi/v3/fw/resource-usage
Fw Server	/axapi/v3/fw/server
Fw Service Group	/axapi/v3/fw/service-group
Fw Status	/axapi/v3/fw/status
Fw System Status	/axapi/v3/fw/system-status
Fw Tap Monitor	/axapi/v3/fw/tap-monitor
Fw Tcp Mss Clamp	/axapi/v3/fw/tcp/mss-clamp
Fw Tcp Reset On Error	/axapi/v3/fw/tcp/reset-on-error
Fw Tcp Rst Close Immediate	/axapi/v3/fw/tcp-rst-close-immediate
Fw Tcp Window Check	/axapi/v3/fw/tcp-window-check
Fw Template Logging	/axapi/v3/fw/template/logging
Fw Top K Rules	/axapi/v3/fw/top-k-rules
Fw Urpf	/axapi/v3/fw/urpf
Fw Vrid	/axapi/v3/fw/vrid
Glm	/axapi/v3/glm
Glm Send	/axapi/v3/glm/send
Harmony Controller Profile	/axapi/v3/harmony-controller/profile
Health Monitor	/axapi/v3/health/monitor
Hostname	/axapi/v3/hostname
Interface Ethernet	/axapi/v3/interface/ethernet
Interface Ethernet Bfd	/axapi/v3/interface/ethernet/{ethernet-ifnum}/bfd
Interface Ethernet Ip	/axapi/v3/interface/ethernet/{ethernet-ifnum}/ip

Display Name	ACOS-AXAPI URI
Interface Ethernet Ipv6	/axapi/v3/interface/ethernet/{ethernet-ifnum}/ipv6
Interface Ethernet Lldp	/axapi/v3/interface/ethernet/{ethernet-ifnum}/lldp
Interface Ethernet Trunk Group	/axapi/v3/interface/ethernet/{ethernet-ifnum}/trunk-group
Interface Lif	/axapi/v3/interface/lif
Interface Lif Ip	/axapi/v3/interface/lif/{lif-ifname}/ip
Interface Loopback	/axapi/v3/interface/loopback
Interface Management	/axapi/v3/interface/management
Interface Ve	/axapi/v3/interface/ve
Interface Ve Bfd	/axapi/v3/interface/ve/{ve-ifnum}/bfd
Interface Ve Ip	/axapi/v3/interface/ve/{ve-ifnum}/ip
Interface Ve Ipv6	/axapi/v3/interface/ve/{ve-ifnum}/ip6
Ip Access List	/axapi/v3/ip/access-list
Ip Address	/axapi/v3/ip/address
Ip Dns Primary	/axapi/v3/ip/dns/primary
Ip Dns Secondary	/axapi/v3/ip/dns/secondary
Ip Dns Suffix	/axapi/v3/ip/dns/suffix
Ip Frag	/axapi/v3/ip/frag
Ip Icmp	/axapi/v3/ip/icmp
Ip Nat Alg Pptp	/axapi/v3/ip/nat/alg/pptp
Ip Nat Global	/axapi/v3/ip/nat-global
Ip Nat Icmp	/axapi/v3/ip/nat/icmp
Ip Nat Pool	/axapi/v3/ip/nat/pool
Ip Prefix List	/axapi/v3/ip/prefix-list
Ip Reroute	/axapi/v3/ip/reroute
Ip Route Rib	/axapi/v3/ipv6/route/rib

Display Name	ACOS-AXAPI URI
Ip Route Static Bfd	/axapi/v3/ipv6/route/static/bfd
Ip Tcp	/axapi/v3/ip/tcp
Ipv6 Frag	/axapi/v3/ipv6/frag
Ipv6 Icmpv6	/axapi/v3/ipv6/icmpv6
Ipv6 Nat Icmpv6	/axapi/v3/ipv6/nat/icmpv6
Ipv6 Nat Pool	/axapi/v3/ipv6/nat/pool
Ipv6 Route Rib	/axapi/v3/ipv6/route/rib
Ipv6 Route Static Bfd Bfd Ipv6	/axapi/v3/ipv6/route/static/bfd/bfd-ipv6
Ipv6 Route Static Bfd Ethernet	/axapi/v3/ipv6/route/static/bfd/ethernet
Ipv6 Route Static Bfd Trunk	/axapi/v3/ipv6/route/static/bfd/trunk
Ipv6 Route Static Bfd Ve	/axapi/v3/ipv6/route/static/bfd/ve
Logging Auditlog	/axapi/v3/logging/auditlog
Logging Console	/axapi/v3/logging/console
Logging Host Ipv4addr	/axapi/v3/logging/host/ipv4addr
Logging Host Ipv6addr	/axapi/v3/logging/host/ipv6addr
Logging Host Partition	/axapi/v3/logging/host/partition
Ntp Auth Key	/axapi/v3/ntp/auth-key
Ntp Server Hostname	/axapi/v3/ntp/server/hostname
Ntp Trusted Key	/axapi/v3/ntp/trusted-key
Overlay Tunnel Options	/axapi/v3/overlay-tunnel/options
Overlay Tunnel Vtep	/axapi/v3/overlay-tunnel/vtep
Partition	/axapi/v3/partition
Reboot	/axapi/v3/reboot
Rib Route	/axapi/v3/reboot
Route Map	/axapi/v3/router/ipv6/rip/route-map
Router Bgp	/axapi/v3/router/bgp
Router Bgp Address Family Ipv6	/axapi/v3/router/bgp/{bgp-as-number}/address-family/ipv6

Display Name	ACOS-AXAPI URI
Router Bgp Address Family Ipv6 Neighbor Ethernet Neighbor Ipv6	/axapi/v3/router/bgp/{bgp-as-number}/address-family/ipv6/neighbor/ethernet-neighbor-ipv6
Router Bgp Address Family Ipv6 Neighbor Ipv4 Neighbor	/axapi/v3/router/bgp/{bgp-as-number}/address-family/ipv6/neighbor/ipv4-neighbor
Router Bgp Address Family Ipv6 Neighbor Ipv6 Neighbor	/axapi/v3/router/bgp/{bgp-as-number}/address-family/ipv6/neighbor/ipv6-neighbor
Router Bgp Address Family Ipv6 Neighbor Peer Group Neighbor	/axapi/v3/router/bgp/{bgp-as-number}/address-family/ipv6/neighbor/peer-group-neighbor
Router Bgp Address Family Ipv6 Neighbor Trunk Neighbor Ipv6	/axapi/v3/router/bgp/{bgp-as-number}/address-family/ipv6/neighbor/trunk-neighbor-ipv6
Router Bgp Address Family Ipv6 Neighbor Ve Neighbor Ipv6	/axapi/v3/router/bgp/{bgp-as-number}/address-family/ipv6/neighbor/ve-neighbor-ipv6
Router Bgp Address Family Ipv6 Network Ipv6 Network	/axapi/v3/router/bgp/{bgp-as-number}/address-family/ipv6/network/ipv6-network
Router Bgp Address Family Ipv6 Network Synchronization	/axapi/v3/router/bgp/{bgp-as-number}/address-family/ipv6/network/synchronization
Router Bgp Address Family Ipv6 Redistribute	/axapi/v3/router/bgp/{bgp-as-number}/address-family/ipv6/redistribute
Router Bgp Neighbor Ethernet Neighbor	/axapi/v3/router/bgp/{bgp-as-number}/neighbor/ethernet-neighbor
Router Bgp Neighbor Ipv4 Neighbor	/axapi/v3/router/bgp/{bgp-as-number}/neighbor/ipv4-neighbor
Router Bgp Neighbor Ipv6 Neighbor	/axapi/v3/router/bgp/{bgp-as-number}/neighbor/ipv6-neighbor

Display Name	ACOS-AXAPI URI
Router Bgp Neighbor Peer Group Neighbor	/axapi/v3/router/bgp/{bgp-as-number}/neighbor/peer-group-neighbor
Router Bgp Neighbor Trunk Neighbor	/axapi/v3/router/bgp/{bgp-as-number}/neighbor/trunk-neighbor
Router Bgp Neighbor Ve Neighbor	/axapi/v3/router/bgp/{bgp-as-number}/neighbor/ve-neighbor
Router Bgp Network Ip Cidr	/axapi/v3/router/bgp/{bgp-as-number}/network/ip-cidr
Router Bgp Network Synchronization	/axapi/v3/router/bgp/{bgp-as-number}/network/synchronization
Router Bgp Redistribute	/axapi/v3/router/bgp/{bgp-as-number}/redistribute
Router Isis	/axapi/v3/router/isis
Router Ospf	/axapi/v3/router/ospf
Router Ospf Area	/axapi/v3/router/ospf/{ospf-process-id}/area
Router Ospf Default Information	/axapi/v3/router/ospf/{ospf-process-id}/default-information
Router Ospf Redistribute	/axapi/v3/router/ospf/{ospf-process-id}/redistribute
Rule Set	/axapi/v3/rule-set
Server	/axapi/v3/slb/server
Service Group	/axapi/v3/slb/service-group
Slb Aflow	/axapi/v3/slb/aflow
Slb Common	/axapi/v3/slb/common
Slb Common Conn Rate Limit Src Ip	/axapi/v3/slb/common/conn-rate-limit/src-ip
Slb Common Buffer Threshold	/axapi/v3/slb/common
Slb Connection Reuse	/axapi/v3/slb/connection-reuse
Slb Crl Srcip	/axapi/v3/slb/crl-srcip
Slb Dns	/axapi/v3/slb/dns

Display Name	ACOS-AXAPI URI
Slb Dns Cache	/axapi/v3/slb/dns-cache
Slb Dns Response Rate Limiting	/axapi/v3/slb/dns-response-rate-limiting
Slb Fast Http Proxy	/axapi/v3/slb/fast-http-proxy
Slb Fix	/axapi/v3/slb/fix
Slb Ftp Ctl	/axapi/v3/slb/ftp-ctl
Slb Ftp Data	/axapi/v3/slb/ftp-data
Slb Ftp Proxy	/axapi/v3/slb/ftp-proxy
Slb Generic Proxy	/axapi/v3/slb/generic-proxy
Slb Health Gateway	/axapi/v3/slb/health-gateway
Slb Health Stat	/axapi/v3/slb/health-gateway/stats
Slb Http2	/axapi/v3/slb/http2
Slb Http Proxy	/axapi/v3/slb/http-proxy
Slb Hw Compress	/axapi/v3/slb/hw-compress
Slb Icap	/axapi/v3/slb/icap
Slb Icap Http	/axapi/v3/slb/icap_http
Slb Imapproxy	/axapi/v3/slb/imap-proxy
Slb L4	/axapi/v3/slb/l4
Slb L7session	/axapi/v3/slb/l7session
Slb Mlb	/axapi/v3/slb/mlb
Slb Mssql	/axapi/v3/slb/mssql
Slb Mysql	/axapi/v3/slb/mysql
Slb Passthrough	/axapi/v3/slb/passthrough
Slb Perf	/axapi/v3/slb/perf
Slb Persist	/axapi/v3/slb/persist
Slb Player Id Global	/axapi/v3/slb/player-id-global
Slb Pop3 Proxy	/axapi/v3/slb/pop3-proxy
Slb Proxy	/axapi/v3/slb/proxy
Slb Rate Limit Log	/axapi/v3/slb/rate-limit-log

Display Name	ACOS-AXAPI URI
Slb Rc Cache Global	/axapi/v3/slb/rc-cache-global
Slb Usage	/axapi/v3/slb/resource-usage
Slb Server Port	/axapi/v3/slb/service-group/{service-group-name}/member/{member-name}+{member-port}
Slb Sip	/axapi/v3/slb/sip
Slb Smpp	/axapi/v3/slb/smpp
Slb Smtp	/axapi/v3/slb/smtp
Slb Spdy Proxy	/axapi/v3/slb/spdy-proxy
Slb Sport Rate Limit	/axapi/v3/slb/sport-rate-limit
Slb Ssl Cert Revoke	/axapi/v3/slb/ssl-cert-revoke
Slb Ssl Expire Check	/axapi/v3/slb/ssl-expire-check
Slb Ssl Forward Proxy	/axapi/v3/slb/ssl-forward-proxy
Slb Svm Source Nat	/axapi/v3/slb/svm-source-nat
Slb Switch	/axapi/v3/slb/switch
Slb Template Cache	/axapi/v3/slb/template/cache
Slb Template Cipher	/axapi/v3/slb/template/cipher
Slb Template Client Ssh	/axapi/v3/slb/template/client-ssh
Slb Template Client Ssl	/axapi/v3/slb/template/client-ssl
Slb Template Connection Reuse	/axapi/v3/slb/template/connection-reuse
Slb Template Csv	/axapi/v3/gslb/template/csv
Slb Template Dblb	/axapi/v3/slb/template/dblb
Slb Template Diameter	/axapi/v3/slb/template/diameter
Slb Template Dns	/axapi/v3/slb/template/dns
Slb Template Dns Class List	/axapi/v3/slb/template/dns/{dns-name}/class-list
Slb Template Dns Logging	/axapi/v3/slb/template/dns-logging
Slb Template Dynamic Service	/axapi/v3/slb/template/dynamic-service

Display Name	ACOS-AXAPI URI
Slb Template External Service	/axapi/v3/slb/template/external-service
Slb Template Fix	/axapi/v3/slb/template/fix
Slb Template Ftp	/axapi/v3/slb/template/ftp
Slb Template Http	/axapi/v3/slb/template/http
Slb Template Http Policy	/axapi/v3/slb/template/http-policy
Slb Template Imap Pop3	/axapi/v3/slb/template/imap-pop3
Slb Template Logging	/axapi/v3/slb/template/logging
Slb Template Monitor	/axapi/v3/slb/template/monitor
Slb Template Mqtt	/axapi/v3/slb/template/mqtt
Slb Template Persist Cookie	/axapi/v3/slb/template/persist/cookie
Slb Template Persist Source Ip	/axapi/v3/slb/template/persist/source-ip
Slb Template Policy	/axapi/v3/slb/template/policy
Slb Template Port	/axapi/v3/slb/template/port
Slb Template Reqmod Icap	/axapi/v3/slb/template/reqmod-icap
Slb Template Respmod Icap	/axapi/v3/slb/template/respmod-icap
Slb Template Server	/axapi/v3/slb/template/server
Slb Template Server Ssh	/axapi/v3/slb/template/server-ssh
Slb Template Server Ssl	/axapi/v3/slb/template/server-ssl
Slb Template Sip	/axapi/v3/slb/template/sip
Slb Template Smpp	/axapi/v3/slb/template/smpp
Slb Template Smtp	/axapi/v3/slb/template/smtp
Slb Template Snmp	/axapi/v3/slb/template/snmp
Slb Template Ssli	/axapi/v3/slb/template/ssli
Slb Template Tcp	/axapi/v3/slb/template/tcp
Slb Template Tcp Proxy	/axapi/v3/slb/template/tcp-proxy
Slb Template Udp	/axapi/v3/slb/template/udp
Slb Template Virtual Port	/axapi/v3/slb/template/virtual-port
Slb Template Virtual Server	/axapi/v3/slb/template/virtual-server

Display Name	ACOS-AXAPI URI
Slb Transparent Acl Template	/axapi/v3/slb/transparent-acl-template
Slb Transparent Tcp Template	/axapi/v3/slb/transparent-tcp-template
Slb Virtual Server Port	/axapi/v3/slb/virtual-server/{virtual-server-name}/port
Snmp Server Contact	/axapi/v3/snmp-server/contact
Snmp Server Disable Traps	/axapi/v3/snmp-server/disable/traps
Snmp Server Enable Traps	/axapi/v3/snmp-server/enable/traps
Snmp Server Enable Traps Gslb	/axapi/v3/snmp-server/enable/traps/gslb
Snmp Server Enable Traps Lsn	/axapi/v3/snmp-server/enable/traps/lsn
Snmp Server Enable Traps Network	/axapi/v3/snmp-server/enable/traps/network
Snmp Server Enable Traps Routing Bgp	/axapi/v3/snmp-server/enable/traps/routing/bgp
Snmp Server Enable Traps Routing Isis	/axapi/v3/snmp-server/enable/traps/routing/isis
Snmp Server Enable Traps Routing Ospf	/axapi/v3/snmp-server/enable/traps/routing/ospf
Snmp Server Enable Traps Slb	/axapi/v3/snmp-server/enable/traps/slb
Snmp Server Enable Traps Slb Change	/axapi/v3/snmp-server/enable/traps/slb-change
Snmp Server Enable Traps Snmp	/axapi/v3/snmp-server/enable/traps/snmp
Snmp Server Enable Traps Ssl	/axapi/v3/snmp-server/enable/traps/ssl
Snmp Server Enable Traps System	/axapi/v3/snmp-server/enable/traps/system
Snmp Server Enable Traps Vcs	/axapi/v3/snmp-server/enable/traps/vcs
Snmp Server Enable Traps Vrrp A	/axapi/v3/snmp-server/enable/traps/vrrp-a
Snmp Server Engine Id	/axapi/v3/snmp-server/engineID
Snmp Server Group	/axapi/v3/snmp-server/group

Display Name	ACOS-AXAPI URI
Snmp Server Host Host Name	/axapi/v3/snmp-server/host/host-name
Snmp Server Host Ipv4 Host	/axapi/v3/snmp-server/host/ipv4-host
Snmp Server Host Ipv6 Host	/axapi/v3/snmp-server/host/ipv6-host
Snmp Server Location	/axapi/v3/snmp-server/location
Snmp Server Management Index	/axapi/v3/snmp-server/management-index
Snmp Server Slb Data Cache Timeout	/axapi/v3/snmp-server/slb-data-cache-timeout
Snmp Server Snmpv1 V2c User	/axapi/v3/snmp-server/SNMPv1-v2c/user
Snmp Server Snmpv1 V2c User Oid	/axapi/v3/snmp-server/SNMPv1-v2c/user/{user-user}/oid
Snmp Server Snmpv3 User	/axapi/v3/snmp-server/SNMPv3/user
Snmp Server User	/axapi/v3/snmp-server/user
Snmp Server View	/axapi/v3/snmp-server/view
System	/axapi/v3/system
System Ve Mac Scheme	/axapi/v3/system/ve-mac-scheme
Timezone	/axapi/v3/timezone
Virtual Server	/axapi/v3/rrd/slb-virtual-server
Vrrp A Vrid	/axapi/v3/vrrp-a/vrid
Vrrp Common	/axapi/v3/vrrp-a/common
Vrrp Peer Group	/axapi/v3/vrrp-a/peer-group
Vrrp Session Sync	/axapi/v3/vrrp-a/session-sync
Web Category	/axapi/v3/web-category
Web Category Category List	/axapi/v3/web-category/category-list
Web Category Proxy Server	/axapi/v3/web-category/proxy-server
Web Category Reputation Scope	/axapi/v3/web-category/reputation-scope
Web Category Statistics	/axapi/v3/web-category/statistics
Write Memory	/axapi/v3/write/memory
Gslb Policy	/axapi/v3/gslb/policy

Display Name	ACOS-AXAPI URI
Gslb Zone	/axapi/v3/gslb/zone
Terminal	/axapi/v3/terminal
Admin Password	/axapi/v3/admin/{admin-user}/password
Gslb Service Ip	/axapi/v3/gslb/service-ip
Gslb Group	/axapi/v3/debug/gslb/group
Gslb Protocol	/axapi/v3/gslb/protocol
Gslb Site	/axapi/v3/gslb/site
System Cpu Ctrl Cpu Oper	/axapi/v3/system-cpu/ctrl-cpu/oper
System Cpu Data Cpu Oper	/axapi/v3/system-cpu/data-cpu/oper

Installing and Initializing Terraform

This section describes the steps to install, initialize and configure Terraform.

The following topics are covered:

Installing Terraform

To install Terraform, download the appropriate package as a zip archive from https://www.terraform.io/downloads. Unzip the downloaded package. Terraform runs as a single binary named **terraform**.

Make sure to include the terraform binary path to the PATH variable. The binary path may differ depending on the operating system.

Initializing Thunder Terraform Provider

After installing Terraform, create a .cfg file as shown below:

version.cfg

```
terraform {
  required_providers {
    thunder = {
      source  = "a10networks/thunder"
      version = "1.1.0" //Change it to the actual version
    }
  }
}
```

Execute **`terraform init`** in the same directory where **`version.cfg`** is placed. Following is the output after executing **`terraform init`** successfully:

```
$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding a10networks/thunder versions matching "1.1.0"...
- Installing a10networks/thunder v0.5.21-beta...
- Installed a10networks/thunder v0.5.21-beta (signed by a HashiCorp
partner, key ID F192222783C8DB3D)

Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it
here:
https://www.terraform.io/docs/cli/plugins/signing.html

Terraform has created a lock file .terraform.lock.hcl to record the
provider
selections it made above. Include this file in your version control
repository
so that Terraform can guarantee to make the same selections by default
when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to
see
any changes that are required for your infrastructure. All Terraform
commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget,
```

```
other
commands will detect it and remind you to do so if necessary.
```

Configuring Thunder Ethernet Interfaces

Create a terraform script file (example **`test.tf`**) as shown below:

`test.tf`:

```
provider "thunder" {
    address  = "x.x.x.x" //your device IP
    username = "x"       //your admin account
    password = "x"       //password of admin account
}

resource "thunder_interface_ethernet" "eth1" {
        ifnum = 1
        action = "enable"
        ip {
                address_list {
                        ipv4_address = "10.101.0.1"
                        ipv4_netmask = "/24"
                }
                address_list {
                        ipv4_address = "10.101.0.81"
                        ipv4_netmask = "/24"
                }
                address_list {
                        ipv4_address = "10.102.0.1"
                        ipv4_netmask = "/24"
                }
        }
}

resource "thunder_interface_ethernet" "eth7" {
        ifnum = 7
        action = "enable"
```

```
        ip {
                address_list {
                        ipv4_address = "10.107.0.1"
                        ipv4_netmask = "/24"
                }
                address_list {
                        ipv4_address = "10.107.0.18"
                        ipv4_netmask = "/24"
                }
                address_list {
                        ipv4_address = "10.108.0.1"
                        ipv4_netmask = "/24"
                }
        }
}
```

Apply the configuration by executing **`terraform apply`** as shown below:

```
$ terraform apply

Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # thunder_interface_ethernet.eth1 will be created
  + resource "thunder_interface_ethernet" "eth1" {
      + action = "enable"
      + id     = (known after apply)
      + ifnum  = 1

      + ip {
          + address_list {
              + ipv4_address = "10.101.0.1"
              + ipv4_netmask = "/24"
            }
          + address_list {
```

```
              + ipv4_address = "10.101.0.81"
              + ipv4_netmask = "/24"
            }
          + address_list {
              + ipv4_address = "10.102.0.1"
              + ipv4_netmask = "/24"
            }
        }
    }

  # thunder_interface_ethernet.eth7 will be created
  + resource "thunder_interface_ethernet" "eth7" {
      + action = "enable"
      + id     = (known after apply)
      + ifnum  = 7

      + ip {
          + address_list {
              + ipv4_address = "10.107.0.1"
              + ipv4_netmask = "/24"
            }
          + address_list {
              + ipv4_address = "10.107.0.18"
              + ipv4_netmask = "/24"
            }
          + address_list {
              + ipv4_address = "10.108.0.1"
              + ipv4_netmask = "/24"
            }
        }
    }

Plan: 2 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.
```

```
  Enter a value: yes

thunder_interface_ethernet.eth1: Creating...
thunder_interface_ethernet.eth7: Creating...
thunder_interface_ethernet.eth7: Creation complete after 3s [id=7]
thunder_interface_ethernet.eth1: Creation complete after 4s [id=1]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

Setting Up HTTPS Virtual Service

Create a terraform script file (example **`test_https.tf`**) as shown below:

`test_https.tf`:

```
provider "thunder" {
    address  = "x.x.x.x"
    username = "x"
    password = "x"
}

//client-side
resource "thunder_interface_ethernet" "eth1" {
    ifnum = 1
    action = "enable"
    ip {
        address_list {
            ipv4_address = "10.101.151.1"
            ipv4_netmask = "/16"
        }
    }
}

//server-side
resource "thunder_interface_ethernet" "eth8" {
    ifnum = 8
    action = "enable"
    ip {
        address_list {
            ipv4_address = "10.102.151.1"
            ipv4_netmask = "/16"
        }
    }
}

resource "thunder_file_ssl_cert" "cert_1" {
    name = "cert_1"
    protocol = "http"
    host = "192.168.92.200"
    path = "/ssl/test/foobar.cert"
    use_mgmt_port = 1
}
```

```
resource "thunder_file_ssl_key" "cert_1_key" {
    name = "cert_1_key"
    protocol = "http"
    host = "192.168.92.200"
    path = "/ssl/test/foobar_nopass.key"
    use_mgmt_port = 1
}

resource "thunder_slb_template_client_ssl" "client_ssl_1" {
    name = "client_ssl_1"
    certificate_list {
        cert = thunder_file_ssl_cert.cert_1.name
        key = thunder_file_ssl_key.cert_1_key.name
    }
}

resource "thunder_server" "server1" {
    name = "server1"
    host = "10.102.156.128"
    port_list {
        port_number = 80
        protocol = "tcp"
    }
}

resource "thunder_service_group" "sg1" {
    name = "sg1"
    protocol = "tcp"
    member_list {
        name = thunder_server.server1.name
        port = 80
    }
}

resource "thunder_virtual_server" "vs1" {
    name = "vs1"
    ip_address = "10.101.151.80"
    port_list {
```

```
        auto = 1 //source-nat
        port_number = 443
        protocol = "https"
        service_group = thunder_service_group.sg1.name
        template_client_ssl = thunder_slb_template_client_ssl.client_ssl_
1.name
    }
}
```

Apply the configuration by executing **`terraform apply`** as shown below:

```
$ terraform apply --auto-approve

Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # thunder_file_ssl_cert.cert_1 will be created
  + resource "thunder_file_ssl_cert" "cert_1" {
      + certificate_type = "pem"
      + host             = "192.168.92.200"
      + id               = (known after apply)
      + name             = "cert_1"
      + overwrite        = 0
      + path             = "/ssl/test/foobar.cert"
      + protocol         = "http"
      + secured          = 0
      + use_mgmt_port    = 1
    }

  # thunder_file_ssl_key.cert_1_key will be created
  + resource "thunder_file_ssl_key" "cert_1_key" {
      + host          = "192.168.92.200"
      + id            = (known after apply)
      + name          = "cert_1_key"
      + overwrite     = 0
      + path          = "/ssl/test/foobar_nopass.key"
      + protocol      = "http"
      + secured       = 0
      + use_mgmt_port = 1
    }

  # thunder_interface_ethernet.eth1 will be created
  + resource "thunder_interface_ethernet" "eth1" {
      + action = "enable"
      + id     = (known after apply)
      + ifnum  = 1
```

```
    + ip {
        + address_list {
            + ipv4_address = "10.101.151.1"
            + ipv4_netmask = "/16"
          }
      }
  }

# thunder_interface_ethernet.eth8 will be created
+ resource "thunder_interface_ethernet" "eth8" {
    + action = "enable"
    + id     = (known after apply)
    + ifnum  = 8

    + ip {
        + address_list {
            + ipv4_address = "10.102.151.1"
            + ipv4_netmask = "/16"
          }
      }
  }

# thunder_server.server1 will be created
+ resource "thunder_server" "server1" {
    + host = "10.102.156.128"
    + id   = (known after apply)
    + name = "server1"

    + port_list {
        + port_number = 80
        + protocol    = "tcp"
      }
  }

# thunder_service_group.sg1 will be created
+ resource "thunder_service_group" "sg1" {
    + id       = (known after apply)
```

```
      + name     = "sg1"
      + protocol = "tcp"

      + member_list {
          + name = "server1"
          + port = 80
        }
    }

  # thunder_slb_template_client_ssl.client_ssl_1 will be created
  + resource "thunder_slb_template_client_ssl" "client_ssl_1" {
      + id   = (known after apply)
      + name = "client_ssl_1"

      + certificate_list {
          + cert = "cert_1"
          + key  = "cert_1_key"
        }
    }

  # thunder_virtual_server.vs1 will be created
  + resource "thunder_virtual_server" "vs1" {
      + id         = (known after apply)
      + ip_address = "10.101.151.80"
      + name       = "vs1"

      + port_list {
          + auto               = 1
          + port_number        = 443
          + protocol           = "https"
          + service_group      = "sg1"
          + template_client_ssl = "client_ssl_1"
        }
    }

Plan: 8 to add, 0 to change, 0 to destroy.
thunder_server.server1: Creating...
thunder_file_ssl_cert.cert_1: Creating...
```

```
thunder_file_ssl_key.cert_1_key: Creating...
thunder_interface_ethernet.eth8: Creating...
thunder_interface_ethernet.eth1: Creating...
thunder_server.server1: Creation complete after 1s [id=server1]
thunder_file_ssl_key.cert_1_key: Creation complete after 1s [id=cert_1_
key]
thunder_service_group.sg1: Creating...
thunder_file_ssl_cert.cert_1: Creation complete after 1s [id=cert_1]
thunder_interface_ethernet.eth1: Creation complete after 1s [id=1]
thunder_slb_template_client_ssl.client_ssl_1: Creating...
thunder_interface_ethernet.eth8: Creation complete after 1s [id=8]
thunder_service_group.sg1: Creation complete after 0s [id=sg1]
thunder_slb_template_client_ssl.client_ssl_1: Creation complete after 0s
[id=client_ssl_1]
thunder_virtual_server.vs1: Creating...
thunder_virtual_server.vs1: Creation complete after 0s [id=vs1]

Apply complete! Resources: 8 added, 0 changed, 0 destroyed.
```

Troubleshooting

In case an error occurs after executing the **`terraform apply`** command, you need to check if any error messages are displayed in the logs.

An error message is displayed in the example below, which shows configuring an object that does not exist:

```
thunder_server.server1: Creating...
thunder_server.server1: Creation complete after 0s [id=server1]
thunder_slb_server_port.svr_port_1: Creating...
╷
│ Error: axapi failure:CM:err=Object specified does not exist (object:
no-such-template)
│
│   with thunder_slb_server_port.svr_port_1,
│   on main.tf line 12, in resource "thunder_slb_server_port" "svr_port_
1":
│   12: resource "thunder_slb_server_port" "svr_port_1" {
```

Full log messages can also be checked on the Thunder device. An example is mentioned below:

```
DEMO>show audit
Feb 25 2022 16:09:04  [admin] cli: [192.168.98.151:61650] show audit
Feb 25 2022 16:02:46  [admin] axapi: [75:192.168.92.193:52286] RESP HTTP
status 404  Not Found : Object specified does not exist
Feb 25 2022 16:02:46  [admin] axapi: [75:192.168.92.193:52286] payload
section 1
{"port":{"port-number":80,"protocol":"tcp","template-server-ssl":"no-
such-template"}}
Feb 25 2022 16:02:46  [admin] axapi: [75:192.168.92.193:52286] POST:
/axapi/v3/slb/server/server1/port
```

Support

You can contact A10 Networks Technical Support by either phone or email. For more information, see A10 Networks Support site.

Additionally, you can also raise feature requests, and report defects using GitHub.

While reporting defects, make sure to include the Terraform script that throws errors and the command output. Including the stack traces can also be very helpful.

NOTE: While raising a defect or sending a feedback, do not include any sensitive information as the **Issues** and **Pull Requests** are publicly viewable.

References

- https://registry.terraform.io/providers/a10networks/thunder/latest
- https://github.com/a10networks/terraform-provider-thunder/branches
- https://documentation.a10networks.com/index.html

Contact Us