Solution Guide for Citrix NetScaler and Cisco APIC EM

Orchestrating Network QoS policy for delivering enhanced video experience to XenDesktop users

CITRIX®

Table of contents

Citrix NetScaler Application Delivery Controller™ (ADC) leverages the Cisco Application Policy Infrastructure Controller (APIC) Enterprise Module's (APIC EM) REST-based APIs to deliver dynamically the best experience for video delivery to Citrix XenDesktop Clients. The APIC EM brings network abstraction and automation for WAN and Access network domains.

Solution Overview

This joint solution between Citrix® NetScaler®, XenDesktop® and Cisco APIC-EM can be used across wired, wireless, physical and virtual networks. This solution protects investment by working with existing infrastructure and it creates an intelligent, open, programmable network that allows the following:

- Quick response to growing application needs.
- Reduces time spent on system configuration.
- Eases the complexity of mobility, bring-your-own-device (BYOD), cloud, and other initiatives.

Executive Summary

Citrix and Cisco jointly deliver a solution for dynamic provisioning of QoS settings across multiple Cisco campus and WAN devices. Upon recognition of XenDesktop Clients that require higher priority for video traffic, the Citrix NetScaler communicates with the Cisco ACI APIC Enterprise Module using Representational State Transfer (REST)-based APIs. In response, the APIC EM pushes new QoS settings to the network devices through which the video traffic passes. Not only are the QoS settings provisioned automatically, it is done on a per session basis, something that is impossible with normal network management tools. The result – an improved XenDesktop user experience.

Challenge

Adapting networks to the security, path optimization, QoS and policy compliance demands of today's applications is complex. A distinct and particular problem is implementing and maintaining network settings across multiple devices in a campus or WAN network. At best, the settings are static and time consuming to implement; at worst, manually implemented, error-prone and poorly managed, or just not feasible.

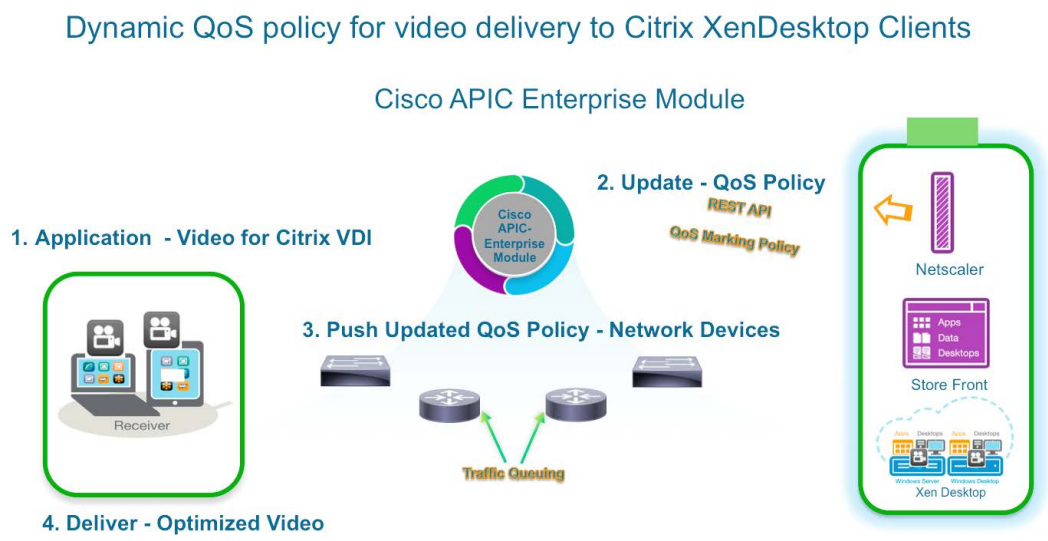
Business Benefits

NetScaler, in conjunction with Cisco APIC EM, can dynamically provision new network parameters in response to changing conditions. This means that systems can be reconfigured more rapidly. This allows better use to be made of existing assets and can lower associated operating expenditure on IT. Further, this flexibility of networking and application provisioning leads to a faster time to deployment of new business applications that can mean a distinct competitive advantage in evolving markets.

Solution Description

To meet the requirements of delivering dynamic QoS policies across campus switches for optimized delivery of video traffic over Citrix XenDesktop Clients, the APIC EM CampusController was installed on a VMware vSphere ESXi host and deployed on a server attached to a Cisco Catalyst 3750-X switch. Citrix XenServer® 6.2 with XenCenter® was installed on a Dell 1435 server and an HP Proliant DL 360-G6 server. The servers were connected to the Cisco Catalyst 3750-X switch. Citrix XenDesktop Windows Server, Citrix NetScaler, and StoreFront virtual environments were deployed on each of those servers.

The APIC-EM CampusController automatically discovered the Citrix XenDesktop solution for video delivery. Once device and host discovery of all the components of the APIC-EM system had been successfully completed, a video was accessed with Citrix Receiver™ client.



The NetScaler device detects the XenDesktop client's action and upon recognizing the data transfer as video determines that a higher prioritization is required. The NetScaler communicates this to the Cisco ACI APIC Enterprise Module using the system's API commands. In response, the APIC EM pushed new QoS settings to the network devices in the data path.

Quality of service provisioning for real-time video delivery is crucial to businesses looking to make the most of a highly diverse and mobile workforce. Without QoS the user experience can be degraded and productivity can suffer. That said, implementing it manually one box at a time can be a challenge for even the most experienced network engineer. The Citrix NetScaler and APIC EM solution can be used to automate QoS provisioning and even create a "follow me" QoS model to establish a consistent, high-quality user experience.

Solution Components and Deployment Details

Physical Components

The APIC-EM Solution is comprised of the following physical components:

1. APIC EM CampusController Server [HP Proliant DL 360-G6 server (ns-tme-ESXi-1)]

2. APIC EM EFT Release 0.7.1.3193

    - Hosted on a VMware vSphere ESXi multipurpose host infrastructure
    - Controls core APIC-EM device inventory and host inventory assets

Cisco Catalyst 3750-X switch (Cat3K)

1. Software version is 15.0(2.0.93)SE

2. Software image is C3750E-UNIVERSALK9-M

3. Switch model number is WS-C3750X-24P

All other physical components of the APIC-EM Demo are connected to this switch for connectivity.

Dell 1435 server (ns-tme-xd-1)

1. XenServer 6.2 with XenCenter

    - RDP Client - This RDP client is used as a jump box to enter the demo

2. NS1 – Citrix NetScaler Virtual Appliance, This Citrix NetScaler pushes the QoS policy to the APIC-EM upon user login

3. Windows Server 2012 – Active Directory

4. Windows Server 2012 – StoreFront1

5. Windows Server 2012 – XenDesktop1

HP Proliant DL 360-G6 server (ns-tme-xd-2)

1. XenServer 6.2 with XenCenter

2. NS2 – Citrix NetScaler Virtual Appliance - This Citrix NetScaler is redundant to the deployment

3. Windows Server 2012 – StoreFront2

4. Windows Server 2012 – XenDesktop2

Virtual Components

The XenDesktop virtual environments installed as virtual machines on the Dell 1435 server are the primary virtual components:

1. RDP Client

2. Citrix NetScaler Virtual Appliance

3. Windows Server 2012

4. Windows Server 2012 – StoreFront1

5. The APIC-EM CampusController

Deployment Details

1. The APIC EM demo was deployed using the following approach.

2. The APIC-EM CampusController was installed on a VMware vSphere ESXi host and deployed on a server attached to the Cisco Catalyst 3750-X switch.

3. XenServer 6.2 with XenCenter was installed on a Dell 1435 server and an HP Proliant DL 360-G6 server.

4. Those servers were attached to the Cisco Catalyst 3750-X switch.

5. The XenDesktop Windows Server, Citrix NetScaler, and StoreFront virtual environments were deployed on each of the servers.

6. The solution environment was brought on line and discovered in the APIC-EM CampusController.

7. Once device discovery and host discovery the physical and virtual components of the APIC-EM had been successfully completed, the solution was ready for launch.

8. A QoS policy was configured on the client port (where the RDP client is attached) of the Cisco Catalyst 3750-X switch using the following command:

   srr-queue bandwidth shape 2 8000 8000 8000

9. QoS policy callouts were configured on the Citrix NetScaler.

   - The Citrix NetScaler pushes the QoS policy to the APIC-EM upon client login and log-out. The QoS policy callouts set the QoS policy upon client login to give the XenDesktop Clients higher priority, and they remove the priority upon client log-out.

10. An RDP client machine that could access the solution to play a video clip was deployed.

11. A "how-to" video was recorded demonstrating the steps necessary to launch the client machine to play the video.
    - The video is first played with no application priority set. In other words, no policy was pushed from the Citrix NetScaler to the APIC-EM.
    - Then, the video was played with a high priority QoS policy pushed from the Citrix NetScaler to the network data path via the APIC-EM.
    - The difference between the two views was noted and quite apparent.

Summary

Citrix and Cisco jointly delivered this differentiated solution because Citrix NetScaler is natively enable with the rich API infrastructure required for seamless integration into the Cisco API-EM. The joint solution between Citrix and Cisco significantly simplifies network change through programmability with an architectural approach and open APIs. Citrix and Cisco services provide end-to-end guidance to help customers achieve maximum benefit from software defined networking and tie network performance directly to business priorities.

For More Information

Citrix NetScaler
http://www.citrix.com/products/netscaler-application-delivery-controller/overview.html?posit=glnav

Citrix XenDesktop
http://www.citrix.com/products/xendesktop/overview.html

Cisco Application Policy Infrastructure Controller Enterprise Module
http://www.cisco.com/c/en/us/products/cloud-systems-management/application-policy-infrastructure-controller-enterprise-module/index.html

Citrix NetScaler and Cisco 3750 Switch Configuration

NetScaler Configuration

```
#NS10.1 Build 123.9
# Last modified by `save config`, Tue Mar 11 21:06:29 2014
set ns config -IPAddress 172.20.239.85 -netmask 255.255.255.128
enable ns feature LB SSL SSLVPN AAA RESPONDER
enable ns mode FR L3 CKA TCPB Edge USNIP PMTUD
set system parameter -natPcbForceFlushLimit 4294967295
set system user nsroot 118c6139d18ce67d70c0dbcb3049826a9b690210fa4eb93a4 -encrypted
set rsskeytype -rsstype ASYMMETRIC
set lacp -sysPriority 32768 -mac 00:25:90:d0:81:8a
set ns hostName NS-DEMO
set interface 0/1 -throughput 0 -bandwidthHigh 0 -bandwidthNormal 0 -intftype “Intel 8247X” -ifnum 0/1
set interface 0/2 -throughput 0 -bandwidthHigh 0 -bandwidthNormal 0 -intftype “Intel 8247X” -ifnum 0/2
set interface 1/1 -throughput 0 -bandwidthHigh 0 -bandwidthNormal 0 -intftype “Intel 8247X” -ifnum 1/1
set interface 1/2 -throughput 0 -bandwidthHigh 0 -bandwidthNormal 0 -intftype “Intel 8247X” -ifnum 1/2
set interface 1/3 -throughput 0 -bandwidthHigh 0 -bandwidthNormal 0 -intftype “Intel 8247X” -ifnum 1/3
set interface 1/4 -throughput 0 -bandwidthHigh 0 -bandwidthNormal 0 -intftype “Intel 8247X” -ifnum 1/4
set interface 1/5 -throughput 0 -bandwidthHigh 0 -bandwidthNormal 0 -intftype “Intel 8247X” -ifnum 1/5
set interface 1/6 -throughput 0 -bandwidthHigh 0 -bandwidthNormal 0 -intftype “Intel 8247X” -ifnum 1/6
set interface LO/1 -haMonitor OFF -throughput 0 -bandwidthHigh 0 -bandwidthNormal 0 -intftype Loopback -ifnum LO/1
```

```
add vlan 2
add vlan 3
add ns ip6 fe80::225:90ff:fed0:818a/64 -scope link-local -type NSIP -vlan 1 -vServer DISABLED
-mgmtAccess ENABLED -dynamicRouting ENABLED
add vrID 11
add vrID 13
add ns ip 30.3.3.12 255.255.255.248 -vServer DISABLED -mgmtAccess ENABLED
add ns ip 192.168.68.247 255.255.255.0 -vServer DISABLED -mgmtAccess ENABLED
add ns ip 172.20.239.114 255.255.255.128 -type MIP -vServer DISABLED
add ns ip 30.3.3.11 255.255.255.255 -type VIP -snmp DISABLED -vrID 11
add ns ip 30.3.3.13 255.255.255.255 -type VIP -snmp DISABLED -vrID 13
set ipsec parameter -lifetime 28800
bind vlan 2 -ifnum 1/2
bind vlan 2 -IPAddress 30.3.3.12 255.255.255.248
bind vlan 3 -ifnum 1/3

bind vlan 3 -IPAddress 192.168.68.247 255.255.255.0
set nd6RAvariables -vlan 1
bind nd6RAvariables -vlan 1 -ipv6Prefix ::
set ipv6 -natprefix ::
set snmp alarm SYNFLOOD -timeout 1
set snmp alarm HA-VERSION-MISMATCH -time 86400 -timeout 86400
set snmp alarm HA-SYNC-FAILURE -time 86400 -timeout 86400
set snmp alarm HA-NO-HEARTBEATS -time 86400 -timeout 86400
set snmp alarm HA-BAD-SECONDARY-STATE -time 86400 -timeout 86400
set snmp alarm HA-PROP-FAILURE -timeout 86400
set snmp alarm IP-CONFLICT -timeout 86400
set snmp alarm APPFW-START-URL -timeout 1
set snmp alarm APPFW-DENY-URL -timeout 1
set snmp alarm APPFW-REFERER-HEADER -timeout 1
set snmp alarm APPFW-CSRF-TAG -timeout 1
set snmp alarm APPFW-COOKIE -timeout 1
set snmp alarm APPFW-FIELD-CONSISTENCY -timeout 1
set snmp alarm APPFW-BUFFER-OVERFLOW -timeout 1
set snmp alarm APPFW-FIELD-FORMAT -timeout 1
set snmp alarm APPFW-SAFE-COMMERCE -timeout 1
set snmp alarm APPFW-SAFE-OBJECT -timeout 1
set snmp alarm APPFW-POLICY-HIT -timeout 1
set snmp alarm APPFW-VIOLATIONS-TYPE -timeout 1
set snmp alarm APPFW-XSS -timeout 1
set snmp alarm APPFW-XML-XSS -timeout 1
set snmp alarm APPFW-SQL -timeout 1
```

```
set snmp alarm APPFW-XML-SQL -timeout 1
set snmp alarm APPFW-XML-ATTACHMENT -timeout 1
set snmp alarm APPFW-XML-DOS -timeout 1
set snmp alarm APPFW-XML-VALIDATION -timeout 1
set snmp alarm APPFW-XML-WSI -timeout 1
set snmp alarm APPFW-XML-SCHEMA-COMPILE -timeout 1
set snmp alarm APPFW-XML-SOAP-FAULT -timeout 1
set snmp alarm DNSKEY-EXPIRY -timeout 1
set snmp alarm HA-LICENSE-MISMATCH -timeout 86400
set snmp alarm CLUSTER-NODE-HEALTH -time 86400 -timeout 86400
set snmp alarm CLUSTER-NODE-QUORUM -time 86400 -timeout 86400
set snmp alarm CLUSTER-VERSION-MISMATCH -time 86400 -timeout 86400
set ns tcpProfile nstcp_default_tcp_lfp -mss 0
set ns tcpProfile nstcp_default_tcp_lnp -mss 0
set ns tcpProfile nstcp_default_tcp_lan -mss 0
set ns tcpProfile nstcp_default_tcp_lfp_thin_stream -mss 0
set ns tcpProfile nstcp_default_tcp_lnp_thin_stream -mss 0

set ns tcpProfile nstcp_default_tcp_lan_thin_stream -mss 0
set ns tcpProfile nstcp_default_tcp_interactive_stream -mss 0
set ns tcpProfile nstcp_internal_apps -mss 0
set ns tcpProfile nstcp_default_XA_XD_profile -mss 0
set ns tcpProfile nstcp_default_Mobile_profile -mss 0
set ns httpProfile nshttp_default_strict_validation -reusePoolTimeout 24820
add policy httpCallout apicall_for_srcpolicy
add policy httpCallout apicall_for_dstpolicy
set policy httpCallout apicall_for_srcpolicy -IPAddress 172.20.239.108 -port 8081 -returnType
TEXT -httpMethod POST -hostExpr “\”172.20.239.108\”” -urlStemExpr “\”/api/v0/policy/\””
-headers Callout(“apicallout”) Content-Type(“application/json”) -bodyExpr “\”{\\x22networkUser\\
x22: {\\x22userIdentifiers\\x22: [\\x2230.3.3.11\\x22]}, \\x22policyName\\x22: \\x22src-citrix\\
x22, \\x22actionProperty\\x22: {\\x22priorityLevel\\x22: \\x2246\\x22}, \\x22policyOwner\\x22:
\\x22Admin\\x22, \\x22actions\\x22: [\\x22PERMIT\\x22]}\”” -scheme http -resultExpr “HTTP.RES.
BODY(100)”
set policy httpCallout apicall_for_dstpolicy -IPAddress 172.20.239.108 -port 8081 -returnType
TEXT -httpMethod POST -hostExpr “\”172.20.239.108\”” -urlStemExpr “\”/api/v0/policy/\””
-headers Callout(“apicallout”) Content-Type(“application/json”) -bodyExpr “\”{\\x22resource\\
x22: {\\x22userIdentifiers\\x22: [\\x2230.3.3.11\\x22]}, \\x22policyName\\x22: \\x22dst-citrix\\
x22, \\x22actionProperty\\x22: {\\x22priorityLevel\\x22: \\x2246\\x22}, \\x22policyOwner\\x22:
\\x22Admin\\x22, \\x22actions\\x22: [\\x22PERMIT\\x22]}\”” -scheme http -resultExpr “HTTP.RES.
BODY(100)”
add server 127.0.0.1 127.0.0.1
add server srv_web 192.168.68.250
```

```
add service s1 127.0.0.1 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0 -cip ENABLED cip-
header -usip YES -useproxyport NO -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO
-TCPB NO -CMP NO -tcpProfileName nstcp_internal_apps
add service svc_web srv_web HTTP 8080 -gslb NONE -maxClient 0 -maxReq 0 -cip
DISABLED -usip NO -useproxyport YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA YES
-TCPB YES -CMP NO
add ssl certKey testcert -cert server.crt -key server.key
add ssl certKey testcert2 -cert server.pem
add ssl certKey serverCA2 -cert ca.pem
add ssl certKey ciscolive -cert ciscolive.cert -key ciscolive.key
add ssl certKey ciscoliveRoot -cert ciscolive-root.cert
add ssl certKey Cybertrust-Ca -cert "Cybertrust Public SureServer SV CA.CER"
add ssl certKey Baltimore-ca -cert "Baltimore CyberTrust Root.CER"
add ssl certKey appflow-cert -cert appflowtrans.citrix.com.CER -key appflowtrans.citrix.
com_pvt.pem
add authentication ldapAction logininvsi -serverIP 192.168.68.245 -ldapBase
"DC=loginvsi,DC=citrix,DC=com" -ldapBindDn administrator@loginvsi.citrix.com
-ldapBindDnPassword dd201c4f799d7879a0c835 -encrypted -ldapLoginName
samAccountName -groupAttrName memberOf -subAttributeName CN -passwdChange
ENABLED
bind cmp global ns_adv_nocmp_xml_ie -priority 8700 -gotoPriorityExpression END -type
RES_DEFAULT
bind cmp global ns_adv_nocmp_mozilla_47 -priority 8800 -gotoPriorityExpression END
-type RES_DEFAULT
bind cmp global ns_adv_cmp_mscss -priority 8900 -gotoPriorityExpression END -type
RES_DEFAULT
bind cmp global ns_adv_cmp_msapp -priority 9000 -gotoPriorityExpression END -type
RES_DEFAULT
bind cmp global ns_adv_cmp_content_type -priority 10000 -gotoPriorityExpression END
-type RES_DEFAULT
add authentication ldapPolicy loginvsiPol ns_true logininvsi
set lb parameter -sessionsThreshold 150000
add lb vserver test_server HTTP 30.3.3.11 80 -persistenceType NONE -cltTimeout 180
add lb vserver internet HTTP 30.3.3.13 80 -persistenceType NONE -cltTimeout 180
set cache parameter -via "NS-CACHE-9.3:   1" -maxPostLen 0
set aaa parameter -maxAAAUsers 5
add vpn vserver ciscodemo SSL 30.3.3.11 443 -appflowLog ENABLED -maxLoginAttempts
255 -failedLoginTimeout 1
add vpn vserver cag_internal SSL 192.168.68.242 443
set ns rpcNode 172.20.239.85 -password
8a7b474124957776a0cd31b862cbe4d72b5cbd59868a136d4bdeb56cf03b28 -encrypted
-srcIP *
add responder policy invoke_src_policy "!http.REQ.HEADER(\"Callout\").EXISTS &&
```

```
http.REQ.URL.CONTAINS(\"LaunchICA\") && SYS.HTTP_CALLOUT(apicall_for_srcpolicy).
CONTAINS(\"success\")" NOOP
add responder policy invoke_dst_policy "!http.REQ.HEADER(\"Callout\").EXISTS &&  http.
REQ.URL.CONTAINS(\"LaunchICA\") && SYS.HTTP_CALLOUT(apicall_for_dstpolicy).
CONTAINS(\"success\")" NOOP
bind responder global invoke_src_policy 1 2 -type REQ_OVERRIDE
bind responder global invoke_dst_policy 2 END -type REQ_OVERRIDE
set responder param -undefAction NOOP
bind lb vserver test_server svc_web
bind lb vserver internet svc_web
add dns nameServer 192.168.68.245
set ns diameter -identity netscaler.com -realm com
set dns parameter -dns64Timeout 1000
add dns nsRec . a.root-servers.net -TTL 3600000
add dns nsRec . b.root-servers.net -TTL 3600000
add dns nsRec . c.root-servers.net -TTL 3600000
add dns nsRec . d.root-servers.net -TTL 3600000

add dns nsRec . e.root-servers.net -TTL 3600000
add dns nsRec . f.root-servers.net -TTL 3600000
add dns nsRec . g.root-servers.net -TTL 3600000
add dns nsRec . h.root-servers.net -TTL 3600000
add dns nsRec . i.root-servers.net -TTL 3600000
add dns nsRec . j.root-servers.net -TTL 3600000
add dns nsRec . k.root-servers.net -TTL 3600000
add dns nsRec . l.root-servers.net -TTL 3600000
add dns nsRec . m.root-servers.net -TTL 3600000
add dns addRec l.root-servers.net 199.7.83.42 -TTL 3600000
add dns addRec b.root-servers.net 192.228.79.201 -TTL 3600000
add dns addRec d.root-servers.net 128.8.10.90 -TTL 3600000
add dns addRec j.root-servers.net 192.58.128.30 -TTL 3600000
add dns addRec h.root-servers.net 128.63.2.53 -TTL 3600000
add dns addRec f.root-servers.net 192.5.5.241 -TTL 3600000
add dns addRec ddc-demo 192.168.68.246
add dns addRec vda-demo 192.168.68.243
add dns addRec rds-demo 192.168.68.244
add dns addRec k.root-servers.net 193.0.14.129 -TTL 3600000
add dns addRec a.root-servers.net 198.41.0.4 -TTL 3600000
add dns addRec c.root-servers.net 192.33.4.12 -TTL 3600000
add dns addRec m.root-servers.net 202.12.27.33 -TTL 3600000
add dns addRec i.root-servers.net 192.36.148.17 -TTL 3600000
add dns addRec g.root-servers.net 192.112.36.4 -TTL 3600000
```

add dns addRec e.root-servers.net 192.203.230.10 -TTL 3600000
set lb monitor ldns-dns LDNS-DNS -query . -queryType Address
add route 0.0.0.0 0.0.0.0 172.20.239.1
add route 30.0.0.0 255.0.0.0 30.3.3.9
set ssl service nshttps-192.168.68.247-443 -eRSA ENABLED -sessReuse DISABLED
set ssl service nsrpcs-192.168.68.247-3008 -eRSA ENABLED -sessReuse DISABLED
set ssl service nshttps-30.3.3.12-443 -eRSA ENABLED -sessReuse DISABLED
set ssl service nsrpcs-30.3.3.12-3008 -eRSA ENABLED -sessReuse DISABLED
set ssl service nshttps-::1l-443 -eRSA ENABLED -sessReuse DISABLED
set ssl service nsrpcs-::1l-3008 -eRSA ENABLED -sessReuse DISABLED
set ssl service nskrpcs-127.0.0.1-3009 -eRSA ENABLED -sessReuse DISABLED
set ssl service nshttps-127.0.0.1-443 -eRSA ENABLED -sessReuse DISABLED
set ssl service nsrpcs-127.0.0.1-3008 -eRSA ENABLED -sessReuse DISABLED
set ssl vserver ciscodemo -eRSA DISABLED -tls11 DISABLED -tls12 DISABLED
set ssl vserver cag_internal -eRSA DISABLED -tls11 DISABLED -tls12 DISABLED
add vpn sessionAction WI-ica-loginvsi -defaultAuthorizationAction ALLOW -SSO ON -icaProxy ON -wihome “http://192.168.68.246/Citrix/ciscoliveWeb/” -ClientChoices ON -ntDomain loginvsi -clientlessVpnMode ON

add vpn sessionPolicy loginvsi_session_pol ns_true WI-ica-loginvsi
set vpn parameter -defaultAuthorizationAction ALLOW -proxy BROWSER -forceCleanup none -clientOptions all -clientConfiguration all -SSO ON -icaProxy ON -wihome “http://192.168.68.246/Citrix/ciscoliveWeb/” -wiPortalMode NORMAL -ClientChoices ON -ntDomain loginvsi clientlessVpnMode ON -UITHEME DEFAULT
bind vpn vserver ciscodemo -staServer “http://192.168.68.246/”
bind vpn vserver ciscodemo -policy loginvsiPol
bind vpn vserver ciscodemo -policy loginvsi_session_pol
bind vpn vserver cag_internal -policy loginvsiPol
bind vpn vserver cag_internal -policy loginvsi_session_pol
bind ssl vserver ciscodemo -certkeyName appflow-cert
bind ssl vserver ciscodemo -certkeyName ciscoliveRoot -CA -ocspCheck Optional
bind ssl vserver ciscodemo -certkeyName Cybertrust-Ca -CA -ocspCheck Optional
bind ssl vserver ciscodemo -certkeyName Baltimore-ca -CA -ocspCheck Optional
bind ssl vserver cag_internal -certkeyName appflow-cert
bind ssl vserver cag_internal -certkeyName ciscoliveRoot -CA -ocspCheck Optional
bind ssl vserver cag_internal -certkeyName Cybertrust-Ca -CA -ocspCheck Optional
bind ssl vserver cag_internal -certkeyName Baltimore-ca -CA -ocspCheck Optional
set ns encryptionParams -method AES256 -keyValue ff0e316156e61a35d08b49fc3083a8a3220ea6019dd2404703d0f050e2004a07c0aaa24b0ecf2d438b2ec8538e1614b97af712b8 – encrypted
set inatparam -nat46v6Prefix ::/96
set ip6TunnelParam -srcIP ::
set ptp -state ENABLE

Citrix NetScaler policies pushed to Cisco APIC EM

The configuration to call the callouts is in the ns.conf and part of the ns gateway config, responder policy takes care of them.

```
add ns variable apic_taskid_map -type “map(text(20),text(64),1000)” -ifValueTooBig undef
-ifNoValue undef
add ns variable apic_policyid_map -type “map(text(20),text(64),1000)” -ifValueTooBig undef
-ifNoValue undef
add ns assignment apic_push_qos_var_assignment -variable “$apic_taskid_map[client.IP.SRC.
TYPECAST_TEXT_T]” -set “sys.HTTP_CALLOUT(apic_push_qos_callout)”
add ns assignment apic_get_qos_var_assignment -variable “$apic_policyid_map[client.IP.SRC.
TYPECAST_TEXT_T]” -set “sys.HTTP_CALLOUT(apic_get_qos_callout)”
```

Login calls apic_push_qos_callout
Logout calls apic_get_qos_callout which then calls apic_delete_qos_callout

```
add policy httpCallout apic_push_qos_callout -vServer apic-em-internal -returnType
TEXT -httpMethod POST -hostExpr 192.168.1.5 -urlStemExpr “\”/api/v0/policy\”” -headers
Content-Type(“application/json”) -bodyExpr q/”{\”actions\”: [\”PERMIT\”], \”policyName\”:
\”Remark-Traffic-Flow\”, \”policyOwner\”: \”admin\”, \”actionProperty\”: {\”priorityLevel\”: \”46\”},
\”networkUser\”: {\”userIdentifiers\”: [\””+CLIENT.IP.SRC.TYPECAST_TEXT_T+”\”]}, \”resource\”:
{\”userIdentifiers\”: [\””+client.IP.DST.TYPECAST_TEXT_T+”\”]}}”/ -scheme http -resultExpr “http.
RES.BODY(10000).REGEX_SELECT(re#\”taskId\”:\”(\\S+)\”#).REGEX_SELECT(re#\\w{8}-\\w{4}-
\\w{4}-\\w{4}-\\w{12}#)”
```

```
add policy httpCallout apic_get_qos_callout -vServer apic-em-internal -returnType TEXT
-hostExpr 192.168.1.5 -urlStemExpr “\”/api/v0/task/\”+$apic_taskid_map[client.IP.SRC.
typecast_text_t]” -headers Accept(“application/json”) -scheme http -resultExpr “http.RES.
BODY(10000).REGEX_SELECT(re#\”progress\”:\”(\\S+)\”#).REGEX_SELECT(re#\\w{8}-\\w{4}-
\\w{4}-\\w{4}-\\w{12}#)”
add policy httpCallout apic_delete_qos_callout -vServer apic-em-internal -returnType TEXT
-fullReqExpr q{“DELETE /api/v0/policy/” + $apic_policyid_map[client.IP.SRC.typecast_text_t] +
“ HTTP/1.1\r\nHost:192.168.1.5\r\nAccept: application/json\r\n\r\n”} -scheme http -resultExpr
“http.RES.BODY(10000)”
set policy httpCallout apic_push_qos_callout -vServer apic-em-internal -returnType TEXT
-httpMethod POST -hostExpr 192.168.1.5 -urlStemExpr “\”/api/v0/policy\”” -headers Content-
Type(“application/json”) -bodyExpr q/”{\”actions\”: [\”PERMIT\”], \”policyName\”: \”Remark-
Traffic-Flow\”, \”policyOwner\”: \”admin\”, \”actionProperty\”: {\”priorityLevel\”: \”46\”},
\”networkUser\”: {\”userIdentifiers\”: [\””+CLIENT.IP.SRC.TYPECAST_TEXT_T+”\”]}, \”resource\”:
{\”userIdentifiers\”: [\””+client.IP.DST.TYPECAST_TEXT_T+”\”]}}”/ -scheme http -resultExpr “http.
RES.BODY(10000).REGEX_SELECT(re#\”taskId\”:\”(\\S+)\”#).REGEX_SELECT(re#\\w{8}-\\w{4}-
\\w{4}-\\w{4}-\\w{12}#)”
set policy httpCallout apic_get_qos_callout -vServer apic-em-internal -returnType TEXT
-hostExpr 192.168.1.5 -urlStemExpr “\”/api/v0/task/\”+$apic_taskid_map[client.IP.SRC.
```

```
typecast_text_t]" -headers Accept("application/json") -scheme http -resultExpr "http.RES.
BODY(10000).REGEX_SELECT(re#\"progress\":\"(\\S+)\"#).REGEX_SELECT(re#\\w{8}-\\w{4}-\\w{4}-
\\w{4}-\\w{12}#)"
set policy httpCallout apic_delete_qos_callout -vServer apic-em-internal -returnType TEXT
-fullReqExpr q{"DELETE /api/v0/policy/" + $apic_policyid_map[client.IP.SRC.typecast_text_t] +
" HTTP/1.1\r\nHost:192.168.1.5\r\nAccept: application/json\r\n\r\n"} -scheme http -resultExpr
"http.RES.BODY(10000)"
```

Cisco 3750 Switch Configuration

```
Cat3K#sh run
Building configuration...

Current configuration : 4705 bytes
!
! Last configuration change at 23:07:13 UTC Tue Mar 16 1993
!
version 15.0
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Cat3K
!
boot-start-marker
boot-end-marker
!
enable password admin
!
username admin password 0 admin
no aaa new-model
switch 1 provision ws-c3750x-24p
system mtu routing 1500
ip routing
!
!
ip device tracking
!
mls qos srr-queue output dscp-map queue 1 threshold 3 32 46
mls qos
!
crypto pki trustpoint TP-self-signed-2833995008
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-2833995008
```

```
 revocation-check none
 rsakeypair TP-self-signed-2833995008
!
!
crypto pki certificate chain TP-self-signed-2833995008
 certificate self-signed 01
  3082022B 30820194 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 32383333 39393530 3038301E 170D3933 30333031 30303031
  33315A17 0D323030 31303130 30303030 305A3031 312F302D 06035504 03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 38333339
  39353030 3830819F 300D0609 2A864886 F70D0101 01050003 818D0030 81890281
  810086F3 F3DDF183 CC96CC86 985AE350 5F6B7396 01302CA3 76EA49B2
633C11A9
  4ADF7968 9FDDD33B 55F6C291 FD6E5D1F C10D8E68 E36EA268 C093363C
F2A33B34
  AC7E3A35 074B16AC AFE0CD73 B9DA03E7 919BD2A0 21811488 99C04A69
AADAD52C
  9CFDA650 4BBA17FB 74725917 CDE21AAF 183B67E1 4D3C2A8F 40DE2E1A
7A59B352
  FE130203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF 301F0603
  551D2304 18301680 146FCA1B 71B78ED2 419D306A E94CFC98 E974C6F6
98301D06
  03551D0E 04160414 6FCA1B71 B78ED241 9D306AE9 4CFC98E9 74C6F698
300D0609
  2A864886 F70D0101 05050003 8181000A 0E985B74 FE3D3683 D249C4C8
385D1B6A
  C981D7A1 F6BF0DA0 E330DFE3 88AD1C9B A25BA56F D98D3F7D F0C6D0D6
D4721A13
  07C501E2 D48F0780 72598691 43882385 ADFD7AA8 08F5C681 62F12D51
F57FF70D
  E78168FB C7BD1A60 F2C5F682 19725635 DC73F438 565D4902 18F3D5F0
A00D7784
  FAFD8F25 1E85893B 50351A8C E58531
      quit
cts role-based enforcement
!
!
!
dot1x system-auth-control
spanning-tree mode pvst
spanning-tree extend system-id
!
```

```
vlan internal allocation policy ascending
!
interface FastEthernet0
 no ip address
 no ip route-cache
!
interface GigabitEthernet1/0/1
!
interface GigabitEthernet1/0/2
!
interface GigabitEthernet1/0/3
!
interface GigabitEthernet1/0/4
!
interface GigabitEthernet1/0/5
!
interface GigabitEthernet1/0/6
!
interface GigabitEthernet1/0/7
!
interface GigabitEthernet1/0/8
!
interface GigabitEthernet1/0/9
!
interface GigabitEthernet1/0/10
 switchport access vlan 100
 switchport mode access
 ip device tracking maximum 1
!
interface GigabitEthernet1/0/11
 description To_Citrix_XD_Server_Farm
 switchport access vlan 100
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 100
 switchport mode trunk
 ip device tracking maximum 10
!
interface GigabitEthernet1/0/12
 description To_Citrix_XD_Server_Farm
 switchport access vlan 100
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 100
 switchport mode trunk
 ip device tracking maximum 10
```

```
!
interface GigabitEthernet1/0/13
 description To_Citrix_XD_Server_Farm
 switchport access vlan 100
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 100
 switchport mode trunk
 ip device tracking maximum 10
 srr-queue bandwidth shape 2 8000 8000 8000
!
interface GigabitEthernet1/0/14
!
interface GigabitEthernet1/0/15
!
interface GigabitEthernet1/0/16
!
interface GigabitEthernet1/0/17
!
interface GigabitEthernet1/0/18
!

interface GigabitEthernet1/0/19
!
interface GigabitEthernet1/0/20
!
interface GigabitEthernet1/0/21
 ip device tracking maximum 1
!
interface GigabitEthernet1/0/22
 ip device tracking maximum 1
!
interface GigabitEthernet1/0/23
 switchport access vlan 2
 switchport mode access
 ip device tracking maximum 10
!
interface GigabitEthernet1/0/24
 switchport mode access
 ip device tracking maximum 10
!
interface GigabitEthernet1/1/1
!
```

```
interface GigabitEthernet1/1/2
!
interface GigabitEthernet1/1/3
!
interface GigabitEthernet1/1/4
!
interface TenGigabitEthernet1/1/1
!
interface TenGigabitEthernet1/1/2
!
interface Vlan1
 ip address 192.168.1.1 255.255.255.0
!
interface Vlan2
 ip address 192.168.2.1 255.255.255.0
!
interface Vlan100
 ip address 10.217.100.205 255.255.255.0
!
ip default-gateway 10.217.100.1

ip http server
ip http secure-server
!
ip sla enable reaction-alerts
!
line con 0
 exec-timeout 0 0
line vty 0 4
 exec-timeout 0 0
 password admin
 login
 transport input telnet
line vty 5 15
 password admin
 login
 transport input telnet
!
end

Cat3K#
```

Cisco Switch Relevant Configuration

show mls qos

config t

mls qos

int gig1/0/12 (config-if)

no mls qos trust dscp

end

show mls qos interface gigabitEthernet 1/0/12

show mls qos interface statistics

show mls qos queue-set 1

show ip device tracking interface gigabitEthernet 1/0/11

show ip device tracking interface gigabitEthernet 1/0/12

show ip device tracking interface gigabitEthernet 1/0/13

config t

int gig1/0/13 (config-if)

no srr-queue bandwidth shape 2 8000 8000 8000

srr-queue bandwidth shape 2 8000 8000 8000

no srr-queue bandwidth shape 2 8000 8000 8000

srr-queue bandwidth shape 2 8000 8000 8000

end

The Command:
srr-queue bandwidth shape 2 8000 8000 8000

Applies the QoS policy to the client port on the switch.

The Command:
no srr-queue bandwidth shape 2 8000 8000 8000

Removes the QoS policy from the client port on the switch.

Corporate Headquarters
Fort Lauderdale, FL, USA

Silicon Valley Headquarters
Santa Clara, CA, USA

EMEA Headquarters
Schaffhausen, Switzerland

India Development Center
Bangalore, India

Online Division Headquarters
Santa Barbara, CA, USA

Pacific Headquarters
Hong Kong, China

Latin America Headquarters
Coral Gables, FL, USA

UK Development Center
Chalfont, United Kingdom

About Citrix
Citrix (NASDAQ:CTXS) is a leader in mobile workspaces, providing virtualization, mobility management, networking and cloud services to enable new ways to work better. Citrix solutions power business mobility through secure, personal workspaces that provide people with instant access to apps, desktops, data and communications on any device, over any network and cloud. This year Citrix is celebrating 25 years of innovation, making IT simpler and people more productive. With annual revenue in 2013 of $2.9 billion, Citrix solutions are in use at more than 330,000 organizations and by over 100 million users globally. Learn more at www.citrix.com.