



FTAS REPORT - Sample SKU X

Execution report generated at: 2024-11-14 14:12:30

Test suite name: /home/oper/testsuites/vendor/data_complete.suite
 Testbed details: ['full_mesh_topo_newtopo.py.html', 'full_mesh_topo_chos.py.html']
 Script running host: ftasvm - 10.20.0.72
 FTAS VERSION: 2.3.1
 SONiC Version: SONiC.Customer.Branch.SONiC.202211
 Execution started: 2024-11-14 12:01:35.595353 - UTC
 Execution completed: 2024-11-14 15:58.11.954313 - UTC
 Execution duration: 3 Hour 57min
 Total TC(s) in Suite 271
 Total TC(s) Executed: 270
 Test case(s) PASSED 258 (95.2%)
 Test case(s) FAILED 12 (4.5%)
 Test case(s) PENDING 1
 Overall result **PASSED**

Test Cases Detailed Logs

ACL IPv4

test_acl_001	Verify L3 DROP ACL functionality with matching source IP and source port	PASSED
test_acl_002	Verify L3 DROP ACL functionality with matching destination IP and destination port	PASSED
test_acl_003	Verify L3 DROP ACL functionality with matching SIP, DIP, SPORT, DPORT	PASSED
test_acl_004	Verify L3 DROP ACL with ACL rule having subnet mask	PASSED
test_acl_005	Verify L3 DROP ACL - Test acl rule with protocol = TCP	PASSED
test_acl_006	Verify L3 DROP ACL - Test acl rule with protocol = UDP	PASSED
test_acl_007	Verify L3 PERMIT ACL functionality with matching source IP and source port	PASSED
test_acl_008	Verify L3 PERMIT ACL functionality with matching destination IP and destination port	PASSED
test_acl_009	Verify L3 PERMIT ACL functionality with matching SIP, DIP, SPORT, DPORT	PASSED
test_acl_010	Verify L3 PERMIT ACL with ACL rule having subnet mask	PASSED
test_acl_011	Verify L3 PERMIT ACL - Test acl rule with protocol = TCP	PASSED
test_acl_012	Verify L3 PERMIT ACL - Test acl rule with protocol = UDP	PASSED

ACL IPv6

test_qual_ip6_acl_001	Verify Drop ACL (IPv6) for matching source IPv6/L4 address and source IPv6L4 port	PASSED
test_qual_ip6_acl_002	Verify Drop ACL (IPv6) for matching destination IPv6/L4 port and source IPv6L4 port	PASSED
test_qual_ip6_acl_003	Verify drop ACL - matching IPv6 params subnet, dst, src ports combined	PASSED
test_qual_ip6_acl_004	Verify PERMIT ACL (IPv6) for matching source IPv6/L4 address and source IPv6L4 port	PASSED
test_qual_ip6_acl_005	Verify Permit ACL (IPv6) for matching destination IPv6/L4 port and source IPv6L4 port	PASSED
test_qual_ip6_acl_006	Verify drop ACL - matching IPv6 params subnet, dst, src ports combined	PASSED

Syslog

test_syslog_002	Verify syslogs are generated properly on link down/up	PASSED
test_syslog_004	Verify syslogs are generated properly on LACP UP/Down	PASSED

Layer 2

test_vlan_014	Verify whether user can configure port as untagged member of a VLAN	PASSED
test_vlan_016	Verify whether known unicast traffic is forwarded to the destination port-channel.	PASSED
test_lacp_003	Verify LACP member addition and removal	PASSED
test_lacp_005	Verify LACP functionality across reboot	PASSED
test_lacp_011	Verify LACP functionality after link failover/failback of physical interface	PASSED
test_lacp_012	Verify LACP functionality after removal and addition of port-channel member	PASSED
test_lldp_001	Enable LLDP globally and disable per-port basis	PASSED
test_lldp_002	Verify that user can enable/disable LLDP globally	PASSED
test_lldp_013	Verify LLDP neighbors are learnt properly with proper ChassisID, portID, system name, system	PASSED
test_arp_003	Verify whether static ARP entry can be configured	PASSED
test_arp_007	Verify that the DUT will respond to an ARP Request for the SVI interface	PASSED
test_arp_011	Verify whether clear ARP entries works properly	PASSED
test_arp_012	Verify whether ARP entries are flushed after some time	PASSED

ARP, IP, Ping

test_arp_011	Verify whether clear ARP entries works properly	PASSED
test_arp_012	Verify whether ARP entries are flushed after some time	PASSED
test_IP_001	Verify that IP address can be configured over SVI	PASSED
test_IP_002	Verify that IP address can be configured over routed port.	PASSED
test_IP_005	Verify SVI and routed ports can be admin down or up	PASSED
test_IP_006	Verify connected route gets created for the SVI subnet in the ip route table.	PASSED
test_IP_011	Verify IP interface is operational for SVI with LACP portchannel members	PASSED
test_IP_014	Verify ip address can be configured over routed PCH.	PASSED
test_ping_001	Verify ping from SONIC SVI interface and routed port	PASSED
test_ping_009	Verify that ping works properly with multiple parameter combination	PASSED
test_ping_011	Verify that ping works properly when using LACP	PASSED
test_ping_013	Verify that ping works over ECMP	PASSED

Port , Platform

test_ports_002	Verify physical port operational down/up	PASSED
test_ports_005	Verify port configuration across reboot	PASSED
test_ports_006	Verify Port Information for status, description and transceiver information	PASSED
test_ports_008	Verify Port Multicast counters	PASSED
test_ports_009_14	Verify Port Counters for framesize 128	PASSED
test_ports_020	Verify Port transceiver information.	PASSED

test_ports_fec_001	Verify FEC Configuration for RS and None	PASSED
test_ports_mtu_002	Verify MTU functionality for Jumboframe packets	PASSED
test_ports_counters	Verify Port Counters for framesize 1518, 9216	PASSED
test_autoneg_001	Verify Ping with port auto-negotiation enabled on both sides	PASSED
test_mtu_001	Verifying IPv4 MTU (1518)	PASSED
test_platform_003	Verify Platform CPU and Process Status	PASSED
test_platform_004	Verify output of show platform psustatus	PASSED
test_platform_005	Verify output of show techsupport	PASSED
test_platform_001	Verify Platform CLI	PASSED
test_platform_002	Verify Platform Health Status	PASSED

test_create_pch_check_rediscli	Create the portchannel and check for the portchannel entry in Asic and LAG_table.	PASSED
test_mem_pch_rediscli_check	Create the portchannel add member port and check their entry in Asic and LAG_table.	PASSED
test_shut_noshut_pch	Create portchannel check shut and no shut functionality on the portchannel.	N/A
test_pch_creation	Create static portchannel and verify traffic flow.	N/A

test_lag_docker_teamd_reboot	Bringup the portchannel and check for its recovery after restarting the teamd container.	N/A
test_pch_sec_member_add_del	Create portchannel add a member ports and check for traffic flow	N/A

Port Breakout

test_port_breakout_001	Dynamic port breakout with supported breakout modes between leaf1 and leaf2	PASSED
------------------------	-----------------------------------------------------------------------------	--------

VLAN

test_vlan_001	Verify whether user can create/delete VLAN	PASSED
test_vlan_002	Verify whether user can add/modify/delete ports to the VLAN as tagged/untagged members	PASSED
test_vlan_004	Verify the ability to configure a port as untagged VLAN member	PASSED
test_vlan_005	Verify the ability to configure a port as tagged VLAN member	PASSED
test_vlan_007	Verify that the user can configure port-channel interface as untagged VLAN member	PASSED
test_vlan_008	Verify that the user can configure port-channel interface as tagged VLAN members	PASSED
test_vlan_011	Verify VLAN over PCH config save and restore after warm restart	PASSED

SSH

test_ssh_001	Verify SSH from host to SONIC on management interface	PASSED
test_ssh_002	Verify SSH from host to SVI interface and routed port	PASSED
test_ssh_003	Verify whether the session is successfully closed right after SSH disconnect from the client.	PASSED

BGP

test_bgp_netops_001	eBGP multi-AS config, adjacency, route convergence and data path using routed interface	PASSED
test_bgp_netops_002	eBGP Multi-AS Route Convergence and data path using loopback	PASSED
test_bgp_netops_003_004	Node drain by applying deny/permit route-maps to remove/restore spine node 1 and 2 using community list	PASSED
test_bgp_netops_005_006	Link drain/restore by applying deny/permit route-map	PASSED
test_bgp_netops_007	Node drain with IPv6 traffic	PASSED
test_bgp_netops_008	Link drain with IPv6 traffic	PASSED
test_bgp_netops_009	Node drain using prefix-lists	PASSED
test_bgp_netops_010	Link drain using prefix-lists	PASSED
test_bgp_netops_011_012	Node drain/restore using AS path prepend list	PASSED
test_bgp_netops_014	Link drain using AS path prepend	PASSED
test_bgp_001	Verify BGP AS configuration works properly	PASSED
test_bgp_002	Verify BGP peering happens with nodes in same AS and iBGP neighbor table gets updated properly	PASSED

test_bgp_003	Verify BGP peering happens with nodes in different AS and eBGP neighbor table gets updated properly	PASSED
test_bgp_004	Verify BGP route learning using eBGP with routes injected from IXIA	PASSED
test_bgp_005	Verify BGP route removal using eBGP with routes withdrawn from IXIA	PASSED
test_bgp_006	Verify BGP route relearn over different neighbor when interface is shutdown	PASSED
test_qual_bgp_001	Verify un numbered functionality with iBGP	PASSED
test_qual_bgp_002	Verify unnumbered functionality with eBGP	PASSED
test_qual_bgp_003	Verify BGP route redistribution in DUT	PASSED
test_qual_bgp_004	Verify BGP6 functionality in DUT	PASSED
test_qual_bgp_007	Verify BGP AS-PATH prepend functionality	PASSED
test_qual_bgp_008	Verify BGP route map match prefix list, access-list deny and permit functionality in DUT	PASSED
test_qual_bgp_009	Verify BGP route map match AS-PATH permit and deny functionality in DUT	PASSED
test_qual_bgp_010	Verify BGP route map match community list permit and deny functionality in DUT	PASSED
test_qual_bgp_011	Verify BGP max MED functionality in DUT	PASSED
test_qual_bgp_013	Verify BGP maximum prefix limit per peer functionality in DUT	FAILED
test_qual_bgp_014	Verify BGP communities functionality in DUT	PASSED
test_qual_bgp_015	Verify BGP regexp match single and multi AS permit & deny action using AS-path access lists	PASSED
test_qual_bgp_016	Verify BGP regexp match any AS permit and deny action using AS-path access lists	PASSED
test_qual_bgp_017	Verify BGP regexp match range of BGP communities functionality in DUT	PASSED
test_qual_bgp_019	Verify BGP peering working with BGP listen range command	PASSED
test_qual_ipv6_neighbor	Verify IPV6 neighbor discovery protocol working	PASSED
test_qual_vrf	Verify VRF functionality	PASSED
test_qual_vlan1	Verify VLAN 1 support for Host connectivity	FAILED
test_bgp_dual_stack_scale_4k	BGP dual stack scale for 4K (IPv4+IPv6) prefix routes	PASSED
test_bgp_dual_stack_scale_16k	BGP dual stack scale for 16K (IPv4+IPv6) prefix routes	PASSED
test_bgp_dual_stack_scale_32k	BGP dual stack scale for 32K (IPv4+IPv6) prefix routes	PASSED
test_bgp_dual_stack_scale_64k	BGP dual stack scale for 64K (IPv4+IPv6) prefix routes	PASSED
test_bgp_dual_stack_scale_128k	BGP dual stack scale for 128K (IPv4+IPv6) prefix routes	PASSED
test_bgp_host_route_dual_stack_32k	BGP dual stack scale with 32K (IPv4+IPv6) host routes with traffic	PASSED
test_bgp_host_route_dual_stack_64k	BGP dual stack scale with 64K (IPv4+IPv6) host routes with traffic	PASSED
test_bgp_host_route_dual_stack_128k	BGP dual stack scale with 128K (IPv4+IPv6) host routes with traffic	PASSED
test_bgp_lpm_route_dual_stack_scale_32k	BGP dual stack scale with 32K (IPv4+IPv6) LPM routes with traffic	PASSED
test_bgp_lpm_route_dual_stack_scale_64k	BGP dual stack scale with 64K (IPv4+IPv6) LPM routes with traffic	PASSED
test_bgp_lpm_route_dual_stack_scale_128k	BGP dual stack scale with 128K (IPv4+IPv6) LPM routes with traffic	PASSED

test_bgp_route_dual_stack_32k	BGP dual stack scale for 32K (IPv4+IPv6) prefix routes (with 16 bit subnet mask for IPv4 and 64bit subnet mask for IPv6 routes) with traffic	PASSED
test_bgp_graceful_restart_preserve_fw_state_32k	BGP graceful restart with dual stack scale to 32K (IPv4+IPv6) prefix routes	PASSED
test_bgp_graceful_restart_preserve_fw_state_64k	BGP graceful restart with dual stack scale to 64K (IPv4+IPv6) prefix routes	PASSED
test_bgp_graceful_restart_preserve_fw_state_128k	BGP graceful restart with dual stack scale to 128K (IPv4+IPv6) prefix routes	PASSED
test_bgp_graceful_docker_restart_32k	BGP graceful restart with dual stack scale to 32K (IPv4+IPv6) prefix routes and docker restart	PASSED
test_bgp_graceful_docker_restart_64k	BGP graceful restart with dual stack scale to 64K (IPv4+IPv6) prefix routes and docker restart	PASSED
test_bgp_graceful_docker_restart_128k	BGP graceful restart with dual stack scale to 128K (IPv4+IPv6) prefix routes and docker restart	PASSED
test_bgp_graceful_feature_disable_enable_32k	Enable/disable BGP graceful restart with dual stack scale to 32K prefix routes	PASSED
test_bgp_kill_process_in_graceful_restart_32k	BGP graceful restart with dual stack scale to 32K (IPv4+IPv6) prefix routes and kill bgp process	PASSED
test_bgp_kill_process_in_graceful_restart_64k	BGP graceful restart with dual stack scale to 64K (IPv4+IPv6) prefix routes and kill bgp process	PASSED
test_bgp_kill_process_in_graceful_restart_128k	BGP graceful restart with dual stack scale to 128K (IPv4+IPv6) prefix routes and kill bgp process	PASSED
test_bgp_dual_stack_scale_intf_flap_160k	BGP interface flap with dual stack scale to 160k(IPv4+IPv6) prefix routes	PASSED
test_bgp_dual_stack_scale_clear_bgp_160k	BGP session down and docker restart with dual stack scale to 160k(IPv4+IPv6) prefix routes	FAILED
test_bgp_dual_stack_convergence_withdrawal_160k	BGP withdraw prefixes(5 times) with dual stack scale 160k(IPv4+IPv6) prefix routes.	PASSED
test_bgp_dynamic_route_scale	Scaling BGP dynamic host routes with traffic.	PASSED
test_bgp_convergence_with_max_med_001	Configure max-med and verify bgp attribute	FAILED
test_bgp_default_route	Default route with bgp advertise routes	PASSED

VxLAN-EVPN

test_bgp4_evpn_vxlan_001	EVPN_VXLAN Configuration and show commands	PASSED
test_bgp4_evpn_vxlan_002	Test EVPN VXLAN for known unicast, BUM traffic (eBGP) with RIF	PASSED
test_bgp4_evpn_vxlan_003	Test EVPN VXLAN for known unicast, BUM traffic (eBGP) with SVI	PASSED
test_bgp4_evpn_vxlan_005	Test EVPN VXLAN for known unicast traffic (eBGP) with link events and router failure - RIF	PASSED
test_bgp4_evpn_vxlan_006	Test EVPN VXLAN for known unicast traffic (eBGP) with link events and router failure - SVI	FAILED
test_bgp4_evpn_vxlan_007	Test EVPN VXLAN for known unicast traffic (eBGP) with link events and router failure - RPCH	FAILED
test_bgp4_evpn_vxlan_009	Test Symmetric IRB with EVPN eBGP-RIF	PASSED
test_bgp4_evpn_vxlan_010	Test Symmetric IRB with EVPN iBGP-RIF	PASSED
test_bgp4_evpn_vxlan_012	Test Symmetric IRB with EVPN iBGP-SVI	PASSED
test_bgp4_evpn_vxlan_013	Test Symmetric IRB with EVPN eBGP-RPCH	PASSED

test_bgp4_evpn_vxlan_014	Test Symmetric IRB with EVPN iBGP-RPCH	PASSED
test_bgp4_evpn_vxlan_015	Test Asymmetric IRB with EVPN eBGP	PASSED
test_bgp4_evpn_vxlan_016	Test Asymmetric IRB with EVPN iBGP	PASSED

SNMP

test_snmp_config	['Verify whether SNMP configurations are working correctly', 'Verify whether SNMP configurations are working correctly', 'Verify whether SNMP configurations are working correctly']	PASSED
test_snmp_commands	Verify SNMP Get/GetNext/Walk requests MIBS- ENTITY, IF-MIB, IP-MIB,	PASSED
test_snmp_walk_inf_admin_oper	Verify that SONiC interface admin and oper status info can be retrieved via SNMP_WALK command	PASSED
test_snmp_walk_inf_index	Verify that SONiC interface index information can be retrieved via SNMP_WALK command	PASSED
test_snmp_walk_inf_type	Verify that SONiC interface type info can be retrieved via SNMP_WALK command	PASSED
test_snmp_walk_ip_inf_index	Verify that SONiC IP interface index and netmask info can be retrieved via SNMP_WALK command	PASSED
test_snmp_walk_ip_to_mac	Verify that IP to MAC (ARP entries) info. can be retrieved via SNMP_WALK command	PASSED
test_snmp_walk_inf_name	Verify that SONiC interface name information can be retrieved via SNMP_WALK command	PASSED
test_snmp_walk_version_serial	Verify that SONiC Version and serial information can be retrieved via SNMP_WALK command	PASSED

Config, Time Zone

test_config_load	Verify Config load for incremental configuration	PASSED
test_front_panel_ports_ipv6	Verify IPv6 address configuration on Front Panel (Data) ports	PASSED
test_cfg_backup_restore	Verify config reload to restore configuration	PASSED
test_timezone_001	Verify timezone can be manually configured.	PASSED
test_ntp_007	Verify NTP server works as clock source correctly	PASSED

SPAN

test_port_span_001	Verify SPAN with source ports (and LAG) to the destination port in ingress/egress/both directions.	PASSED
test_port_span_002	Verify ERSPAN by configuring a mirror with the list of source ports/LAG to destination IP in ingress/egress/both directions	PASSED
test_port_span_003	Verify ERSPAN by configuring a mirror from ACL match to destination IP in ingress directions	PASSED

TACACS+

test_tacacs_001	Verify Tacacs+ with AAA authentication	PASSED
-----------------	----------------------------------------	--------

Storm Control

test_storm_control_cli_verification	Validate Storm Control CLI Configuration and Behavior.	PASSED
-------------------------------------	--------------------------------------------------------	--------

test_storm_control_invalid_input	Verify DUT throws proper error on giving invalid input.	PASSED
test_storm_control_broadcast	Storm Control with Broadcast traffic	FAILED
test_storm_control_unknown_unicast	Storm Control with Unknown-Unicast traffic	FAILED
test_storm_control_unknown_multicast	Storm Control with Unknown-multicast traffic	FAILED
test_storm_control_warm_reboot	Validate DUT Storm Control functionality with warm-reboot	PASSED

Scaling

test_v4_host_routes_scale_2k	IPv4 host routes scale to 2K routes	PASSED
test_v4_host_routes_scale_4k	IPv4 host routes scale to 4K routes	PASSED
test_v4_host_routes_scale_64k	IPv4 host routes scale to 64K routes	PASSED
test_v4_host_routes_scale_max_supported	IPv4 host routes scale to maximum supported routes (MAX_IPV4_HOST_ROUTES)	PASSED
test_v6_host_routes_scale_2k	IPv6 host routes scale to 2K routes (eBGP)	PASSED
test_v6_host_routes_scale_4k	IPv6 host routes scale to 4k route (eBGP)	PASSED
test_v6_host_routes_scale_32k	IPv6 host routes scale to 32k route (eBGP)	PASSED
test_v6_host_routes_scale_64k	IPv6 host routes scale to 64k route (iBGP)	PASSED
test_v6_host_routes_scale_max_supported	IPv6 host routes scale to maximum supported routes (MAX_IPV6_HOST_ROUTES)	PASSED
test_v4_prefix_routes_scale_2k	IPv4 prefix routes scale to 2k routes	PASSED
test_v4_prefix_routes_scale_4k	IPv4 prefix routes scale to 4k routes	PASSED
test_v4_prefix_routes_scale_64k	IPv4 prefix routes scale to 4k routes	PASSED
test_v4_prefix_routes_scale_max_supported	IPv4 prefix routes scale to maximum supported routes (MAX_IPV4_PREFIX_ROUTES)	PASSED
test_v4_nexthops_scale_512	64 IPv4 nexthops with 512 routes	PASSED
test_v4_nexthops_scale_1024	64 IPv4 nexthops with 1024 routes	FAILED
test_v4_nexthops_scale_max_supported	64 IPv4 nexthops with maximum supported routes (MAX_IPV4_ROUTES_PER_NEXTHOP)	FAILED
test_qual_scale_011	Scale up to two syslog servers	FAILED
test_v4_acl_scale_128	128 IPv4 ACL rules (matching source IP/port and destination IP/port)	PASSED
test_v4_acl_scale_256	256 IPv4 ACL rules (matching source IP/port and destination IP/port)	PASSED
test_v4_acl_scale_512	512 IPv4 ACL rules (matching source IP/port and destination IP/port)	PASSED
test_v4_acl_scale_max_supported	(MAX_V4_ACL) IPv4 ACL rules (matching source IP/port and destination IP/port)	PASSED
test_qual_v6_acl_scale_128	128 IPv6 ACL rules (denying source IPv6/port and destination IPv6/port)	PASSED
test_qual_v6_acl_scale_256	256 IPv6 ACL rules (denying source IPv6/port and destination IPv6/port)	PASSED
test_qual_v6_acl_scale_512	512 IPv6 ACL rules (denying source IPv6/port and destination IPv6/port)	PASSED
test_qual_v6_acl_scale_max_supported	(MAX_V6_ACL) IPv6 ACL rules (denying source IPv6/port and destination IPv6/port)	PASSED
test_16ecmp_32k_routes	Scalability to 16 ECMP paths and 32K routes	PASSED
test_32ecmp_64k_routes	Scalability to 32 ECMP paths and 64K routes	PASSED
test_64ecmp_128k_routes	Scalability to 64 ECMP paths and 128K routes	FAILED

test_ecmp_scale_routes	Scalability to "number of device ports" ECMP paths and 1K routes per path	PASSED
test_ebgp_16ecmp_32k_routes	Scalability to 16 ECMP paths and 32K eBGP routes	PASSED
test_ebgp_32ecmp_64k_routes	Scalability to 32 ECMP paths and 64K eBGP routes	PASSED
test_ebgp_64ecmp_128k_routes	Scalability to 64 ECMP paths and 128K eBGP routes	PASSED
test_ebgp_ecmp_scale_routes	Scalability to ECMP paths and 1K eBGP routes per path	PASSED
test_ingress_v4_acl_scale_128	128 IPv4 ACL rules (matching source IP/port and destination IP/port)	PASSED
test_ingress_v4_acl_scale_256	128 IPv4 ACL rules (matching source IP/port and destination IP/port)	PASSED
test_ingress_v4_acl_scale_512	128 IPv4 ACL rules (matching source IP/port and destination IP/port)	PASSED
test_ingress_v4_acl_scale_max_supported	(MAX_V4_ACL) IPv4 ACL rules (matching source IP/port and destination IP/port)	PASSED
test_ipv6_l3_prefix_routes_scale_4k	Scalability to 4K IPv6 prefix routes with traffic.	PASSED
test_ipv6_l3_prefix_routes_scale_64k	Scalability to 64K IPv6 prefix routes with traffic.	PASSED
test_v6_host_routes_scale_ibgp_2k	IPv6 host routes scale to 2K routes (iBGP)	PASSED
test_v6_host_routes_scale_ibgp_32k	IPv6 host routes scale to 32k route (iBGP)	PASSED
test_v6_host_routes_scale_ibgp_4k	IPv6 host routes scale to 4k route (iBGP)	PASSED