



Reference - Basic Monitoring and Advanced Monitoring Rules

This section explains the following topics:

- [Basic and Advanced Monitoring Rules, on page 1](#)

Basic and Advanced Monitoring Rules

Crosswork Service Health monitoring provides two options for monitoring: Basic Monitoring and Advanced Monitoring. The table below outlines the monitoring functions of each rule and sub-services, as well as the metric dependencies for both Basic and Advanced monitoring rules included in the system-defined Heuristic Package:

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
Rule-L2VPN-NM-Basic	<ul style="list-style-type: none">• Checks the health of the VPWS xconnect state.• Monitors the health of the device: CPU and memory utilization.	subservice.device.health subservice.vpws.ctrlplane.health metric.l2vpn.xconnect.state metric.l2vpn.xconnect.ac.state metric.l2vpn.xconnect.pw.state

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
Rule-L2VPN-NM (Advanced)		

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
	<ul style="list-style-type: none"> • Checks the health of the VPWS or EVPN xconnect state. • Monitors the health of the device: CPU and memory utilization. • Monitors the delta between received and transmitted packets between VPN interfaces and Pseudo-wire. • Monitors Y.1731 probe stats for jitter, loss, and delay metrics, and compares against SLA thresholds. • Monitors the health status of RSVP tunnel. Subservice health will be marked as 'degraded' in either of the below scenarios: <ul style="list-style-type: none"> • FRR is configured, but backup is not ready. • FRR backup is active (primary failed and traffic is flowing over FRR backup). • Health check for interface metrics: Oper status, interface in/out error packets, interface in/out packet discard. • Checks BGP Neighbor session health. • Checks whether all BGP EVPN next hops for a given L2VPN service are reachable over LSP. • Monitors PCEP session state to all the peers configured on this device. • Checks path reachability between two endpoints. • SR Policy (PCC initiated) 	subservice.bgp.nbr.health subservice.bgp.evpn.nexthop.health subservice.device.health subservice.evpn.health (one for each endpoint) subservice.fallback.path.health subservice.interface.health (one for each interface) subservice.l2vpn.y1731.health subservice.path.reachability.to.peer (local to remote and remote to local) subservice.path.sla subservice.pcep.session.health (one for each endpoint device) subservice.plain.lsp.path.health subservice.sr.policy.pce.health (one for each endpoint) subservice.vpws.ctrlplane.health (local, remote) subservice.path.reachability.to.peer subservice.fallback.path.health subservice.mpls.rsvpte.tunnel.pm.health subservice.l2vpn.y1731.health subservice.vpws.ctrlplane.health subservice.interface.health subservice.device.health subservice.interface.health.summary subservice.path.sla.summary metric.bgp.router.id metric.cef.route.labeled.lsp metric.l2vpn.xconnect.ac.state metric.l2vpn.xconnect.pw.state metric.l2vpn.xconnect.state metric.device.xconnect.ac.in.packets metric.device.xconnect.pw.out.packet metric.l2vpn.y1731.connect.cross.check.status metric.interface.oper

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
	<p>health status. Admin should be up. Oper should be up. Oper should have stayed up since last polling.</p> <ul style="list-style-type: none"> Checks whether LSP path exists (in default VRF) towards the given destination device. 	<p>metric.interface.in.errors metric.device.cpu.load metric.device.memory.free</p>
Rule-L2VPN-NM-P2P-Basic	<ul style="list-style-type: none"> Checks the health of the VPWS xconnect state. Monitors the health of the device: CPU and memory utilization. 	<p>subservice.device.health subservice.vpws.ctrlplane.health</p>
Rule-L2VPN-NM-P2P (Advanced)	<ul style="list-style-type: none"> Checks the health of the VPWS xconnect state. Monitors the health of the device: CPU and memory utilization. Checks the health for interface metrics: Oper status, interface in/out error packets, interface in/out packet discard. Monitors Y.1731 probe stats for jitter, loss, and delay metrics, and compares against SLA thresholds. Monitors the LSP path to the peer VPN node. Monitors path reachability between two endpoints. Monitors LSP path (in default VRF) towards the given destination IP address. Monitors PCEP session state to all the peers configured on this device. Checks the SR Policy (PCC initiated) health status. Admin should be up. Oper should be up. Oper should have stayed up since last polling. 	<p>subservice.device.health subservice.interface.health (one for each interface) subservice.l2vpn.y1731.health subservice.p2p.fallback.path.health subservice.p2p.path.reachability.to.peer (path reachability between endpoints) subservice.p2p.plain.lsp.path.health subservice.path.sla subservice.pcep.session.health (one for each endpoint device) subservice.sr.policy.pcc.health subservice.sr.policy.pce.health (one for each endpoint) subservice.vpws.ctrlplane.health (local, remote) metric.cef.route.labeled.lsp metric.l2vpn.xconnect.ac.state metric.l2vpn.xconnect.pw.state metric.l2vpn.xconnect.state</p>

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
Rule-L2VPN-MP-Basic	<ul style="list-style-type: none">• For all .summary subservices: Groups together all the device subservices as an aggregator node. It does not have its own health/metric. Its health depends on its child subservice health.• Monitors the health of the device• Monitors bridge domain state on a given endpoint.	subservice.device.summary subservice.bridge.domain.summary subservice.device.health subservice.bridge.domain.state

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
Rule-L2VPN-MP (Advanced)		

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
	<ul style="list-style-type: none"> • For all .summary subservices: Groups together all the device subservices as an aggregator node. It does not have its own health/metric. Its health depends on its child subservice health. • Monitors the health of the device. • Groups together all the PCEP session health subservices. • Monitors PCEP session state to all the peers configured on this device. • Groups together all the device subservices. • Checks BGP Neighbor health. • Monitors whether any routes are present for the given Bridge Domain. • Groups together all the bridge domain subservices. • Monitors bridge domain state on a given endpoint. • Subservice to reflect interface health. • Groups together all the transport subservices. • SR Policy health status reflecting SR-PM SLA (if configured). Admin and Oper should be up. Oper should have stayed up since last polling. Delay and Variance should meet SLA if SR-PM is configured to measure delay. Liveness should be up if SR-PM is configured for Liveness. • Monitors the policies deployed by the ODN. 	<p>subservice.device.summary subservice.device.health subservice.pcep.session.health.summary subservice.pcep.session.health subservice.evpn.summary subservice.bgp.nbr.health subservice.mac.learning subservice.bridge.domain.summary subservice.bridge.domain.state subservice.interface.health subservice.transport.summary subservice.sr.policy.pcc.pm.health subservice.sr.policy.pce.pm.health subservice.mpls.rsvpte.tunnel.pm.health subservice.l2vpn.sr.odn.policy.dynamic metric.device.memory.free (supports XR only) metric.device.cpu.load (supports XR only) metric.sr.te.pcc.peer.state (supports XR only) metric.sr.te.pcc.peer.addrs (supports XR only) metric.bgp.session.state (supports XR only) metric.bgp.neighbors.ipaddr.list (supports XR only) metric.mac.learning.nexthops (supports XR only) metric.l2vpn.bridge.ac.state (supports XR only) metric.l2vpn.bridge.ac.list (supports XR only) metric.l2vpn.bridge.domain.state (supports XR only) metric.interface.oper (supports both XR and XE) metric.interface.in.errors (supports both XR and XE) metric.interface.out.errors (supports both XR and XE) metric.interface.in.discards (supports both XR and XE)</p>

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
	<ul style="list-style-type: none"> • SR Policy health status that include SR-PM. Admin and Oper should be up, and Oper should have stayed up since last polling. Delay and Variance should meet SLA if SR-PM is configured to measure delay. Liveness should be up if SR-PM is configured for Liveness. • Monitors MPLS RSVP TE Tunnel Health. Admin, Oper should both be up and if FRR is configured, then backup path should be ready to pickup traffic when primary fails. If failover already happened to backup then health will be shown as degraded as there is no more redundancy in play. Delay should be considered if SR-PM is enabled. If delay is enabled, then variance will be considered. 	<p>metric.interface.out.discards (supports both XR and XE)</p> <p>metric.sr.policy.pcc.admin.state (supports XR only)</p> <p>metric.sr.policy.pcc.oper.state (supports XR only)</p> <p>metric.sr.policy.pcc.oper.up.time (supports XR only)</p> <p>metric.sr.policy.pm.delay.measurement (supports XR only)</p> <p>metric.sr.pm.delay (supports XR only)</p> <p>metric.sr.pm.variance (supports XR only)</p> <p>metric.sr.policy.pm.liveness.detection (supports XR only)</p> <p>metric.sr.pm.liveness.state (supports XR only)</p> <p>metric.sr.policy.pcc.admin.state (supports XR only)</p> <p>metric.sr.policy.pcc.oper.state (supports XR only)</p> <p>metric.sr.policy.pcc.oper.up.time (supports XR only)</p> <p>metric.sr.policy.pcc.ietf.policy.name (supports XR only)</p> <p>metric.sr.policy.pm.delay.measurement (supports XR only)</p> <p>metric.sr.pm.delay (supports XR only)</p> <p>metric.sr.pm.variance (supports XR only)</p> <p>metric.sr.policy.pm.liveness.detection (supports XR only)</p> <p>metric.sr.pm.liveness.state (supports XR only)</p> <p>metric.mpls.rsvpte.tunnel.oper.state (supports XR only)</p> <p>metric.mpls.rsvpte.tunnel.admin.state (supports XR only)</p> <p>metric.mpls.rsvpte.tunnel.frr.configured (supports XR only)</p> <p>metric.mpls.rsvpte.tunnel.frr.status (supports XR only)</p>

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
		metric.mpls.te.pm.delay.measurement (supports XR only) metric.mpls.rsvp.te.delay (supports XR only) metric.mpls.rsvp.te.variance (supports XR only) metric.l2vpn.odn.sr.policies.list (supports XR only) metric.bgp.router.id (supports both XR and XE)
Rule-L3VPN-NM-Basic	<ul style="list-style-type: none"> • Reports the overall route connectivity health between the current PE device and its connecting CE device. • Monitors the health of the device: CPU and memory utilization. 	subservice.ce.pe.route.health subservice.device.health

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
Rule-L3VPN-NM (Advanced)	<ul style="list-style-type: none"> • For all .summary subservices: Groups together all the device subservices as an aggregator node. It does not have its own health/metric. Its health depends on its child subservice health. • Subservice, together with child subservices in L3VPN Rule, reports the overall route health between current PE device and its connecting CE device. • eBGP Session health • Subservice to reflect interface health. • Monitors the health of the device. • L3VPN Aggregator Subservice that reflects path reachability from given device, for a given vrf, to peer VPN sites. • Monitors both static and dynamically initiated policy. • Checks whether plain LSP route exists within given VRF towards given vpn ip-addresses. • Monitors PCEP session state to all the peers configured on this device. • Checks BGP Neighbor health. 	

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
		subservice.ce.pe.route.health.summary subservice.ce.pe.route.health subservice.ebgp.nbr.health subservice.interface.health.summary subservice.interface.health subservice.device.summary subservice.device.health subservice.vrf.path.reachability.to.peer.summary subservice.vrf.path.reachability.to.peers subservice.transport.summary subservice.dynamic.l3vpn.sr.policy subservice.vrf.plain.lsp.reachability subservice.pcep.session.health.summary subservice.pcep.session.health subservice.bgp.nbr.health.summary subservice.bgp.nbr.health subservice.bgp.evpn.nextthop.health subservice.bgp.nbr.health subservice.ce.pe.route.health subservice.device.health subservice.ebgp.nbr.health subservice.evpn.health subservice.fallback.path.health subservice.interface.health subservice.l2vpn.y1731.health subservice.p2p.fallback.path.health subservice.p2p.path.reachability.to.peer subservice.p2p.plain.lsp.path.health subservice.path.reachability.to.peer subservice.path.sla subservice.pcep.session.health subservice.plain.lsp.path.health subservice.sr.policy.pcc.health

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
		subservice.sr.policy.pce.health subservice.vpws.ctrlplane.health subservice.vrf.path.reachability.to.peers subservice.vrf.plain.lsp.reachability subservice.bridge.domain.summary subservice.l3vpn.sr.odn.policy.dynamic subservice.l2vpn.sr.odn.policy.dynamic subservice.mac.learning subservice.mpls.rsvpte.tunnel.pm.health subservice.vrf.path.reachability.to.peer.summary subservice.path.sla.summary subservice.pcep.session.health.summary subservice.transport.summary subservice.interface.health.summary subservice.vpws.ctrlplane.health.summary subservice.bridge.domain.state metric.route.vrf.connected (supports XR and XR IPv6) metric.route.vrf.local (supports XR and XR IPv6) metric.bgp.vrf.session.state (supports XR only) metric.interface.oper (supports both XR and XE) metric.interface.in.errors (supports both XR and XE) metric.interface.out.errors (supports both XR and XE) metric.interface.in.discards (supports both XR and XE) metric.interface.out.discards (supports both XR and XE) metric.device.memory.free (supports XR only) metric.device.cpu.load (supports XR only) metric.l3vpn.sr.policies.list (supports XR and XR IPv6)

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
		<p>metric.cef.vrf.route.prefix (supports XR and XR IPv6)</p> <p>metric.sr.te.pcc.peer.state (supports XR only)</p> <p>metric.sr.te.pcc.peer.addrs (supports XR only)</p> <p>metric.bgp.session.state (supports XR only)</p> <p>metric.bgp.neighbors.ipaddr.list (supports XR only)</p> <p>metric.bgp.route.l2vpn.evpn.nexthops</p> <p>metric.bgp.router.id</p> <p>metric.cef.route.labeled.lsp</p> <p>metric.bgp.session.state</p> <p>metric.bgp.neighbors.ipaddr.list</p> <p>metric.route.vrf.connected</p> <p>metric.route.vrf.local</p> <p>metric.device.memory.free</p> <p>metric.device.cpu.load</p> <p>metric.bgp.vrf.session.state</p> <p>metric.l2vpn.xconnect.pw.state</p> <p>metric.cef.route.labeled.lsp</p> <p>metric.bgp.router.id</p> <p>metric.interface.oper</p> <p>metric.interface.in.errors</p> <p>metric.interface.out.errors</p> <p>metric.interface.in.discards</p> <p>metric.interface.out.discards</p> <p>metric.l2vpn.y1731.connect.cross.check.status</p> <p>metric.l2vpn.y1731.connect.peer.mep.status</p> <p>metric.l2vpn.y1731.latency.rt</p> <p>metric.l2vpn.y1731.jitter.rt</p> <p>metric.l2vpn.y1731.pktloss.lway.sd</p> <p>metric.l2vpn.y1731.pktloss.lway.ds</p> <p>metric.cef.route.labeled.lsp</p> <p>metric.cef.route.labeled.lsp</p> <p>metric.device.xconnect.ac.in.packets</p>

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
		metric.device.xconnect.pw.out.packets metric.device.xconnect.pw.in.packets metric.device.xconnect.ac.out.packets metric.sr.te.pcc.ipv4.peer.state metric.sr.te.pcc.ipv4.peer.addr metric.cef.route.labeled.lsp metric.bgp.router.id metric.sr.policy.pcc.oper.state metric.sr.policy.pcc.oper.up.time metric.sr.policy.pcc.admin.state metric.sr.policy.pm.delay.measurement metric.sr.pm.delay metric.sr.pm.variance metric.sr.policy.pm.liveness.detection metric.sr.pm.liveness.state metric.sr.policy.pce.oper.up.time metric.sr.policy.pce.oper.state metric.sr.policy.pce.admin.state metric.l2vpn.xconnect.state metric.l2vpn.xconnect.ac.state metric.l2vpn.xconnect.pw.state metric.cef.vrf.route.prefix metric.l3vpn.odn.sr.policies.dynamic.list metric.l2vpn.odn.sr.policies.list metric.bgp.router.id metric.mac.learning.nexthops metric.mpls.rsvpte.tunnel.oper.state metric.mpls.rsvpte.tunnel.admin.state metric.mpls.rsvpte.tunnel.frr.configured metric.mpls.rsvpte.tunnel.frr.status metric.mpls.te.pm.delay.measurement metric.mpls.rsvp.te.delay metric.l2vpn.bridge.ac.state

Rule Name (Type)	Monitoring Functionality	Metrics and Subservices
		metric.l2vpn.bridge.ac.list metric.l2vpn.bridge.domain.state

Example

The given example explains the relationship between the 'Rule-L2VPN-NM-P2P-Basic' and its dependent sub-services, specifically 'subservice.vpws.ctrlplane.health' and 'subservice.device.health'. Additionally, the sub-service definitions are also listed below to highlight the metric dependencies and symptoms generated by these sub-services.

Rule-L2VPN-NM-P2P-Basic

```
{
  "name": "Rule-L2VPN-NM-P2P-Basic",
  "namespace": "system",
  "id": "Rule-L2VPN-NM-P2P-Basic system",
  "description": "Rule to generate Assurance Graph for Basic L2VPN NM P2P Services.",
  "matchCriteria": [
    {
      "configSource": "SOURCE_TYPE_NS0",
      "configSubSource": [
        "SUBSOURCE_SERVICE_CONFIG"
      ],
      "matchType": "MATCH_TYPE_XPATH",
      "matchExpression":
        "//vpn-service[@xmlns='urn:ietf:params:xml:ns:yang:ietf-l2vpn-ntw']/vpn-svc-type[text()='vpn-common:t-ldp']",
      "matchPrefix": "",
      "matchParams": []
    },
    {
      "configSource": "SOURCE_TYPE_NS0",
      "configSubSource": [
        "SUBSOURCE_SERVICE_CONFIG"
      ],
      "matchType": "MATCH_TYPE_XPATH",
      "matchExpression":
        "//flat-L2vpn/service-type[text()='p2p']",
      "matchPrefix": "",
      "matchParams": []
    },
    {
      "configSource": "SOURCE_TYPE_NS0",
      "configSubSource": [
        "SUBSOURCE_SERVICE_CONFIG"
      ],
      "matchType": "MATCH_TYPE_XPATH",
      "matchExpression":
        "//vpn-service[@xmlns='urn:ietf:params:xml:ns:yang:ietf-l2vpn-ntw']/vpn-type[text()='vpn-common:t-ldp']",
      "matchPrefix": "",
      "matchParams": []
    },
    {
      "configSource": "SOURCE_TYPE_NS0",
      "configSubSource": [
        "SUBSOURCE_SERVICE_CONFIG"
      ],
      "matchType": "MATCH_TYPE_XPATH",
```

```

        "matchExpression": "//vpn-service[not(//bridge-group)]/vpn-type[contains(text(),
':mpls-evpn')]",
        "matchPrefix": "",
        "matchParams": []
    },
    {
        "configSource": "SOURCE_TYPE_NS0",
        "configSubSource": [
            "SUBSOURCE_SERVICE_CONFIG"
        ],
        "matchType": "MATCH_TYPE_XPATH",
        "matchExpression":
        "//vpn-service[@xmlns='urn:ietf:params:xml:ns:yang:ietf-l2vpn-ntw']/vpn-type[text()='x:vpws']",

        "matchPrefix": "",
        "matchParams": []
    },
    {
        "configSource": "SOURCE_TYPE_NS0",
        "configSubSource": [
            "SUBSOURCE_SERVICE_CONFIG"
        ],
        "matchType": "MATCH_TYPE_XPATH",
        "matchExpression":
        "//vpn-service[@xmlns='urn:ietf:params:xml:ns:yang:ietf-l2vpn-ntw']/vpn-type[text()='ietf-vpn-common:vpws']",

        "matchPrefix": "",
        "matchParams": []
    }
],
"dependencies": [
    {
        "name": "VPWS-ControlPlane-Health-Summary",
        "id": "subservice.vpws.ctrlplane.health.summary.system",
        "ssClass": "subservice.vpws.ctrlplane.health.summary",
        "namespace": "system",
        "type": "DEP_TYPE_NON_LIST",
        "optional": false,
        "paramExtractionMechanism": {
            "mode": "EXTRACT_MODE_XPATH",
            "name": "",
            "namespace": "",
            "version": "",
            "validationHash": "0",
            "pluginMethod": "",
            "extractedParams": [],
            "nativeMethod": ""
        },
        "parameters": [
            {
                "name": "vpnServiceId",
                "iterator": false,
                "defaultValue": "",
                "extractionMethod": "DEP_PARAM_XPATH",
                "extractionDetails": [
                    {
                        "description": "",
                        "extractValue": "//vpn-service/vpn-id"
                    },
                    {
                        "description": "Flat Model",
                        "extractValue": "//flat-L2vpn[/flat-L2vpn-p2p]/key"
                    }
                ]
            }
        ]
    }
]

```



```

    }
  ],
  "subDependencies": [
    "VPWS-ControlPlane-Health-Local-Site",
    "VPWS-ControlPlane-Health-Remote-Site"
  ],
  "softSubDependencies": []
},
{
  "name": "VPWS-ControlPlane-Health-Local-Site",
  "id": "subservice.vpws.ctrlplane.health system",
  "ssClass": "subservice.vpws.ctrlplane.health",
  "namespace": "system",
  "type": "DEP_TYPE_NON_LIST",
  "optional": false,
  "paramExtractionMechanism": {
    "mode": "EXTRACT_MODE_XPATH",
    "name": "",
    "namespace": "",
    "version": "",
    "validationHash": "0",
    "pluginMethod": "",
    "extractedParams": [],
    "nativeMethod": ""
  },
  "parameters": [
    {
      "name": "device",
      "iterator": false,
      "defaultValue": "",
      "extractionMethod": "DEP_PARAM_XPATH",
      "extractionDetails": [
        {
          "description": "",
          "extractValue": "//vpn-nodes/vpn-node[1]/vpn-node-id"
        }
      ]
    },
    {
      "name": "groupName",
      "iterator": false,
      "defaultValue": "",
      "extractionMethod": "DEP_PARAM_XPATH",
      "extractionDetails": [
        {
          "description": "",
          "extractValue": "//vpn-service/vpn-id"
        },
        {
          "description": "Flat Model",
          "extractValue": "//flat-L2vpn/flat-L2vpn-p2p/local-site/xconnect-group-name"
        }
      ]
    },
    {
      "name": "xconnectName",
      "iterator": false,
      "defaultValue": "",
      "extractionMethod": "DEP_PARAM_XPATH",
      "extractionDetails": [
        {
          "description": "",
          "extractValue": "//vpn-service/vpn-id"
        }
      ]
    }
  ]
}

```

```

        {
          "description": "Flat Model",
          "extractValue": "//flat-L2vpn/flat-L2vpn-p2p/local-site/xconnect-group-name"
        }
      ]
    },
    "subDependencies": [],
    "softSubDependencies": [
      "device1"
    ]
  },
  {
    "name": "VPWS-ControlPlane-Health-Remote-Site",
    "id": "subservice.vpws.ctrlplane.health system",
    "ssClass": "subservice.vpws.ctrlplane.health",
    "namespace": "system",
    "type": "DEP_TYPE_NON_LIST",
    "optional": false,
    "paramExtractionMechanism": {
      "mode": "EXTRACT_MODE_XPATH",
      "name": "",
      "namespace": "",
      "version": "",
      "validationHash": "0",
      "pluginMethod": "",
      "extractedParams": [],
      "nativeMethod": ""
    },
    "parameters": [
      {
        "name": "device",
        "iterator": false,
        "defaultValue": "",
        "extractionMethod": "DEP_PARAM_XPATH",
        "extractionDetails": [
          {
            "description": "",
            "extractValue": "//vpn-nodes/vpn-node[2]/vpn-node-id"
          }
        ]
      },
      {
        "name": "groupName",
        "iterator": false,
        "defaultValue": "",
        "extractionMethod": "DEP_PARAM_XPATH",
        "extractionDetails": [
          {
            "description": "",
            "extractValue": "//vpn-service/vpn-id"
          },
          {
            "description": "Flat Model",
            "extractValue": "//flat-L2vpn/flat-L2vpn-p2p/remote-site/xconnect-group-name"
          }
        ]
      }
    ]
  },
  {
    "name": "xconnectName",
    "iterator": false,
    "defaultValue": "",
    "extractionMethod": "DEP_PARAM_XPATH",

```

```

    "extractionDetails": [
      {
        "description": "",
        "extractValue": "//vpn-service/vpn-id"
      },
      {
        "description": "Flat Model",
        "extractValue": "//flat-L2vpn/flat-L2vpn-p2p/remote-site/xconnect-group-name"
      }
    ]
  },
  "subDependencies": [],
  "softSubDependencies": [
    "device2"
  ]
},
{
  "name": "device1",
  "id": "subservice.device.health system",
  "ssClass": "subservice.device.health",
  "namespace": "system",
  "type": "DEP_TYPE_NON_LIST",
  "optional": false,
  "paramExtractionMechanism": {
    "mode": "EXTRACT_MODE_XPATH",
    "name": "",
    "namespace": "",
    "version": "",
    "validationHash": "0",
    "pluginMethod": "",
    "extractedParams": [],
    "nativeMethod": ""
  },
  "parameters": [
    {
      "name": "device",
      "iterator": false,
      "defaultValue": "",
      "extractionMethod": "DEP_PARAM_XPATH",
      "extractionDetails": [
        {
          "description": "",
          "extractValue": "//vpn-nodes/vpn-node[1]/vpn-node-id"
        }
      ]
    }
  ],
  "subDependencies": [],
  "softSubDependencies": []
},
{
  "name": "device2",
  "id": "subservice.device.health system",
  "ssClass": "subservice.device.health",
  "namespace": "system",
  "type": "DEP_TYPE_NON_LIST",
  "optional": false,
  "paramExtractionMechanism": {
    "mode": "EXTRACT_MODE_XPATH",
    "name": "",
    "namespace": "",
    "version": "",

```

```

        "validationHash": "0",
        "pluginMethod": "",
        "extractedParams": [],
        "nativeMethod": ""
    },
    "parameters": [
        {
            "name": "device",
            "iterator": false,
            "defaultValue": "",
            "extractionMethod": "DEP_PARAM_XPATH",
            "extractionDetails": [
                {
                    "description": "",
                    "extractValue": "//vpn-nodes/vpn-node[2]/vpn-node-id"
                }
            ]
        }
    ],
    "subDependencies": [],
    "softSubDependencies": []
}
},
"softRootDependencies": [],
"createTimestamp": "1697841637567500247",
"updateTimestamp": "0",
"monitoringType": "BASIC",
"version": "1.1"
}

Sub service: 'subservice.vpws.ctrlplane.health'

{
    "id": "subservice.vpws.ctrlplane.health.summary system",
    "name": "subservice.vpws.ctrlplane.health.summary",
    "namespace": "system",
    "description": "Groups together all the VPWS Ctrlplane health subservices.",
    "params": [
        {
            "name": "vpnServiceId",
            "description": "",
            "type": "PARAM_TYPE_NON_LIST"
        }
    ],
    "liveMetrics": {},
    "rootExpressions": [],
    "dynamicConfig": null,
    "symptom": null,
    "dependencies": [],
    "exprCid": "",
    "createTimestamp": "1697841637373426164",
    "updateTimestamp": "0",
    "tags": [],
    "version": "1.0"
}

{
    "id": "subservice.vpws.ctrlplane.health system",
    "name": "subservice.vpws.ctrlplane.health",
    "namespace": "system",
    "description": "check the health of the VPWS state",
    "params": [
        {
            "name": "device",
            "description": "",

```

```

        "type": "PARAM_TYPE_NON_LIST"
    },
    {
        "name": "groupName",
        "description": "",
        "type": "PARAM_TYPE_NON_LIST"
    },
    {
        "name": "xconnectName",
        "description": "",
        "type": "PARAM_TYPE_NON_LIST"
    }
],
"liveMetrics": {},
"rootExpressions": [
    {
        "evalExpression": "xconnect_state == 'up' && ac_state == 'up' && evpn_state == 'up'",
        "activateCondition": ""
    }
],
"dynamicConfig": null,
"symptom": {
    "formatString": "VPWS State degraded. Device: {device}, XConnectGroup: {groupName},
XconnectName: {xconnectName}",
    "level": "DEGRADED",
    "priority": 15,
    "condition": false
},
"dependencies": [
    {
        "type": "DEP_TYPE_METRIC",
        "label": "xconnect_state",
        "evalExpression": "metric.l2vpn.xconnect.state",
        "namespace": "",
        "symptom": null,
        "paramMap": {
            "device": "device",
            "groupName": "groupName",
            "xconnectName": "xconnectName"
        },
        "id": ""
    },
    {
        "type": "DEP_TYPE_METRIC",
        "label": "ac_state",
        "evalExpression": "metric.l2vpn.xconnect.ac.state",
        "namespace": "",
        "symptom": null,
        "paramMap": {
            "device": "device",
            "groupName": "groupName",
            "xconnectName": "xconnectName"
        },
        "id": ""
    },
    {
        "type": "DEP_TYPE_METRIC",
        "label": "evpn_state",
        "evalExpression": "metric.l2vpn.xconnect.pw.state",
        "namespace": "",
        "symptom": null,
        "paramMap": {
            "device": "device",
    
```

```

        "groupName": "groupName",
        "xconnectName": "xconnectName"
    },
    "id": ""
}
],
"exprCid": "",
"createTimestamp": "1697841637370064741",
"updateTimestamp": "0",
"tags": [],
"version": "1.0"
}

```

Sub service: 'subservice.device.health'

```

{
  "id": "subservice.device.health system",
  "name": "subservice.device.health",
  "namespace": "system",
  "description": "Monitor the health of the device.",
  "params": [
    {
      "name": "device",
      "description": "",
      "type": "PARAM_TYPE_NON_LIST"
    }
  ],
  "liveMetrics": {},
  "rootExpressions": [
    {
      "evalExpression": "cpu_healthy && memory_healthy",
      "activateCondition": ""
    }
  ],
  "dynamicConfig": null,
  "symptom": {
    "formatString": "Heavier than expected resource consumption on the Device: {device}",
    "level": "DEGRADED",
    "priority": 100,
    "condition": false
  },
  "dependencies": [
    {
      "type": "DEP_TYPE_EXPRESSION",
      "label": "cpu_healthy",
      "evalExpression": "ListElemsAverage(cpu_load) <= CPU_THRESHOLD_MAX",
      "namespace": "",
      "symptom": null,
      "paramMap": {},
      "id": ""
    },
    {
      "type": "DEP_TYPE_EXPRESSION",
      "label": "memory_healthy",
      "evalExpression": "ListElemsSum(memory_free) > MEMFREE_THRESHOLD_MIN",
      "namespace": "",
      "symptom": null,
      "paramMap": {},
      "id": ""
    },
    {
      "type": "DEP_TYPE_METRIC",
      "label": "cpu_load",
      "evalExpression": "metric.device.cpu.load",
      "namespace": "",

```

```

        "symptom": null,
        "paramMap": {
          "device": "device"
        },
        "id": ""
      },
      {
        "type": "DEP_TYPE_METRIC",
        "label": "memory_free",
        "evalExpression": "metric.device.memory.free",
        "namespace": "",
        "symptom": null,
        "paramMap": {
          "device": "device"
        },
        "id": ""
      }
    ],
    "exprCid": "",
    "createTimestamp": "1697841637256704609",
    "updateTimestamp": "0",
    "tags": [
      "DEVICE_SUBSERVICES"
    ],
    "version": "1.1"
  }

  {
    "id": "subservice.device.summary system",
    "name": "subservice.device.summary",
    "namespace": "system",
    "description": "Groups together all the Device subservices",
    "params": [
      {
        "name": "vpnServiceId",
        "description": "",
        "type": "PARAM_TYPE_NON_LIST"
      }
    ],
    "liveMetrics": {},
    "rootExpressions": [],
    "dynamicConfig": null,
    "symptom": null,
    "dependencies": [],
    "exprCid": "",
    "createTimestamp": "1697841637260108075",
    "updateTimestamp": "0",
    "tags": [],
    "version": "1.0"
  }

```

