

Juniper Networks – AppFormix /TRY Training Script

Revision: 1.5

Date: 14 May 2018

Infrastructure:

Try Contrail and AppFormix Sandbox

<https://www.juniper.net/us/en/cloud-software/trial/index.html>

User Guide

https://www.juniper.net/documentation/en_US/appformix/information-products/pathway-pages/pwp-appformix-reference-guide.html

Product Materials

<https://www.juniper.net/uk/en/products-services/application-management-orchestration/appformix/>

Sandbox Login Instructions:

- Await the 'Your reservation is ready!' email from Juniper Cloud Labs and follow the instructions.
- Use the search facility on your laptop to find the RDP Remote Desktop Connection application
- Use RDP to connect to the IP Address provided in the reservation email and manually type in the Username and Password
- Use Firefox or Chrome on the RDP client window to connect to the Openstack, Contrail and AppFormix dashboards using the bookmarked URLs. The Usernames and Passwords will be provided in the reservation email.
- **Important:** Use a separate browser per dashboard GUI instance to help with in-context navigation using RDP

Sandbox Topology Information:

The /TRY sandbox environment contains an instance of Openstack, Contrail, AppFormix and one compute node for running VM based workloads.

There are three physical hosts used in the sandbox:

- nugget-cfg – this hosts the Openstack and Contrail Controllers
- nugget-appformix – this hosts the AppFormix Controller
- nugget-compute – this is the compute node for the system where VM based workloads can run. The networking layer for this node is provided by the Contrail vRouter

AppFormix Workshop Session Use Cases:

The following set of AppFormix use cases can be run in the /TRY sandbox environment.

1. Infrastructure Visualization, Navigation and Resource Grouping Business value: Higher degrees of visualization and understanding of cloud infrastructure dependencies leads to operation cost reduction of managing complex cloud environments. Aggregation and tagging concepts can be a powerful visualization and automation productivity tool. Differentiation: The creation of Aggregates and Tags enables cloud infrastructure components to be grouped together into a common context and provides the ability to link resources by physical, virtual or service based models. In addition to being a visualization aid to help understand workload and infrastructure dependencies, it enables dynamic SLA policy creation based on pre-defined groupings.			
1a	Navigation and visualization of managed infrastructure	<p>The AppFormix GUI consists of a context-driven infrastructure navigation menu bar across the top horizontal axis of the GUI that enables granular selection of grouped infrastructure resources. The feature tabs on the vertical left-hand side of the GUI enable access to AppFormix capabilities in the context of the top menu bar settings.</p> <p>Test Case:</p> <ol style="list-style-type: none"> 1. Log into the AppFormix Dashboard 2. Start with the menu setting on 'Infrastructure' and the feature tab set to 'Dashboard'. Examine the top level Infrastructure Dashboard screen and the infrastructure layers that are grouped together. <ul style="list-style-type: none"> • The view shows the managed resource counts for each infrastructure layer and a general view of component health 3. Clicking on any of the coloured circular status discs will show incremental information on the resources being monitored and the current health status 4. Navigate down and across the drop down navigation menu items to understand the way the menu bar builds out. <p>Note: most of the feature tabs change the context of what is displayed on the main screen based on the selections of the navigation menu bar.</p>	
1b	Auto-discovered Project groups	Projects are discovered and aligned, where applicable, with the Virtual Infrastructure Management (VIM) layer that AppFormix is integrated with.	

		<p>Test Case:</p> <ol style="list-style-type: none"> 1. Log into the AppFormix and Openstack GUIs 2. Start with the AppFormix menu setting on 'Projects -> Dashboard' and compare with Projects listed under the Identity tab on the Openstack Horizon GUI <ul style="list-style-type: none"> • AppFormix will have discovered the Openstack projects and added a couple of default project groups of its own. 	
1c	Create a tagged based aggregate group	<p>Group related instances into an Aggregate group using the GUI.</p> <p>Test Case:</p> <ol style="list-style-type: none"> 1. Use the AppFormix search facility to find an instance object called 'demo' and select the instance record called 'demo-vm1' 2. Hover the mouse over the three vertical dots under the Tags section of the GUI (top right-hand side of the window) 3. Select '+ New' and add a new aggregate called 'Traffic Test' 4. Navigate to 'demo-vm2' and add it to the same 'Traffic Test' aggregate 5. Click on the 'Traffic Test' aggregate icon and only the two instances will be shown 6. Click on the 'Charts' tab for the for the 'Traffic Test' aggregate and view the composite metrics <p>Note: Large scale management of aggregate groups can be achieved with the use of the API.</p>	Current

2. Analysis of Operational State of Cloud Virtual Infrastructure Management Layer

Business Value: Operation cost reduction of managing cloud management infrastructure resources.

Differentiation: Monitoring the health of the VIM layer and critical software services is an extremely important operational requirement for any cloud. Any performance degradation of the VIM or software service dependencies will lead to cloud service quality level reductions over time.

	Title	Description	Availability
2a	Real-time monitoring of VIM infrastructure availability and performance	<p>AppFormix monitors the health of the VIM it is connected to. In this /TRY case, AppFormix is integrated to an Openstack VIM environment.</p> <p>Test Case:</p> <ol style="list-style-type: none"> 1. Log into the AppFormix Dashboard 2. Visualize the AppFormix 'Infrastructure -> Dashboard' 3. Navigate to Keystone from either the Keystone Widget or via Services on the infrastructure navigation menu bar 4. Examine the Top 5 Clients. They will be the IP Addresses of the Contrail and AppFormix servers that authenticate with Keystone. 5. Examine the API interaction and performance activity in the Status Code Count widget 6. Click on the Historical button to see the past health state of the cluster 7. Look for any risk and bad health status icons 8. Navigate to the 'Nova -> Dashboard' and repeat the analysis of the health of the environment. 	Current
2b	Real-time monitoring of infrastructure services	<p>AppFormix monitors the health of Infrastructure services including Contrail Networking.</p> <p>Test Case:</p> <ol style="list-style-type: none"> 1. Open the Services menu item and examine the list of services that AppFormix supports via plug-ins. Note: the services list can be extended. 2. Navigate to the 'Contrail -> Dashboard' and repeat the analysis of the health of the environment. 3. Check if any Contrail alarm conditions exist and view the health history of the environment. 	Current

		4. Check the status of each of the Contrail Group Nodes and Peer Groups using the navigation menu	
--	--	---	--

3. Analysis of Operational State of Infrastructure Resources

Business Value: Operation cost reduction of navigating the topology of managed infrastructure resources.

Differentiation: Complete visibility for resource metric objects with relationships to physical and virtual infrastructure resource consumption. Enables simple investigations into 'noisy workloads' and the impact on infrastructure consumption.

	Title	Description	Availability
3a	Real-time monitoring of cloud infrastructure availability and performance	<p>AppFormix monitors the health of physical infrastructure hosts.</p> <p>Test Case:</p> <ol style="list-style-type: none"> 1. Log into the AppFormix Dashboard 2. Visualize Hosts on the AppFormix dashboard. Hosts can be accessed from the Infrastructure -> Dashboard Hosts widget or the menu navigation dropdown bar. <ul style="list-style-type: none"> • Examine the health and risk status of each host • Check the host IP Addresses • Confirm which host is supporting VM based instance workloads. 3. Navigate to the Charts tab and check CPU, memory, disk, network and other infrastructure and application metrics collected by AppFormix. The values will be compared across all the hosts in the menu bar context. 4. Select the 'nugget-compute' node from the menu bar and <ul style="list-style-type: none"> - In the Chart tab examine the network interface metrics for physical hosts (example host.network.ingress.packet_rate) - In the Dashboard tab look at the Instances running on the Hosts and navigate down to each instance and then back to the host 5. Look for any risk and Bad Health status icons 	Current

3b	Real-time monitoring of infrastructure modifications	<p>AppFormix monitors the health of Infrastructure as dynamic changes occur and keep aligned in real-time.</p> <p>Test Case:</p> <ol style="list-style-type: none"> 1. Open the Openstack Horizon GUI 2. Find the existing instances under the 'appformix' project. Note: Check you are in the correct project context in the Openstack GUI 3. Add (launch) a new VM instance (select Ubuntu-stress-test instance, m1.medium, demo-vm1.vnet) 4. Navigate to the AppFormix dashboard 5. Use the search facility to find the new VM just added in Openstack 6. Check the Project and confirm the Flavor configurations by clicking on the Flavor label. Note how many vCPUs are allocated. 7. Navigate to the host the VM is sitting on by clicking on the label icon under the Host: field 8. Click on the chart icon under the View: field and examine the report for CPU utilization. The chart will show the CPU utilization figures for all the VMs running on the host. 	Current
3c	AppFormix monitors dynamic changes in infrastructure metrics	<p>Raise the consumption on the resources and servers being monitored by AppFormix Agents:</p> <p>Test Case:</p> <ol style="list-style-type: none"> 1. Log into the newly created VM via the console of the Openstack Horizon GUI (use contrail/contrail123 credentials) 2. Run command to burn CPU. <ul style="list-style-type: none"> • yes > /dev/null & (consumes the first vCPU allocated to the VM) 3. Monitor the CPU usage of the server increase. <ul style="list-style-type: none"> • View the CPU load from an instance perspective • View the CPU load from the underlying host perspective 4. Kill the yes script and check the CPU usage drops <ul style="list-style-type: none"> • killall yes <p>Visualize Health, Risk and SLA profiles indicating health of resources being good, bad or at risk.</p>	Current

		<p>5. Use mprime tests for additional resource stress tests and use the Charts view for metric analysis.</p> <p>Note: The AppFormix agent is configured by default to upload metric statistics to the controller every 60 seconds and carefully scheduled based on the size of the estate to protect cloud resource performance. It is expected that the reports window will take up to 60 seconds to display updated metric readings. However, events generated by the agent are sent instantaneously.</p>	
--	--	---	--

4. Closed-loop Enforcement of Resource Management SLAs

Business Value: Higher degrees of customer SLA leading to incremental service revenue options. Operation cost reduction of managing infrastructure resources by providing near real-time notification of infrastructure health issues.

Differentiation: Real-time state-based compliance events can be used to notify operations staff or send to self-healing automated functions

	Title	Description	Availability
4a	SLA notification for overconsumption of workload resources on instance	<p>AppFormix responsively detects exhaustion of compute resource on an instance.</p> <p>Test Cases:</p> <ol style="list-style-type: none"> 1. Navigate to the Setting menu (top right face icon) 2. Open the SLA Settings tab 3. Configure a Risk SLA rule for instance CPU usage thresholds. <ul style="list-style-type: none"> • Navigate to the Instance tab • Delete the default profile • Add a new rule for 'Instance' CPU exceeding 49% Use the Mode setting as 'Alert' to give the rule a state (i.e. Active, Inactive etc.) Set the Interval Duration to 10 seconds so that the alert condition is created after 10 seconds from the condition change. 	Current

		<ul style="list-style-type: none"> • Trigger the rules on the Ubuntu-stress-test instance with tools like yes > /dev/null & (forces 50% CPU utilization on the VM) <p>4. Validate the SLA instance risk event is raised in AppFormix. Look for the Health Risk icon against the VM instance.</p>	
4b	SLA notification for overconsumption of resources on a Project	<p>AppFormix responsively detects exhaustion of resource in a project.</p> <p>Test Cases:</p> <ol style="list-style-type: none"> 1. Navigate to the Setting menu (top right face icon) 2. Open the SLA Settings tab 3. Configure a Risk SLA rule for 'Project' CPU usage thresholds. <ul style="list-style-type: none"> • Navigate to the Instance tab • Delete the default profile • Add a new for instance CPU exceeding 49% 4. Validate the SLA project risk condition is raised in AppFormix against the 'appformix' project 	Current

5. Alarms and Notifications, Machine Learning Analytics

Business Value: Operations cost reduction through risk and failure detection automation. Risk reduction on service quality breaches and activation failures to sustain and assure customer revenue.

Differentiation: Real-time analysis of cross-layer infrastructure metrics with machine learning and abnormality analytics. The first step of an infrastructure self-healing process.

5a	Alarm notification for overconsumption of infrastructure resources	<p>AppFormix detects an excessive load on a resource. An alarm condition is created with a static threshold that could later be used for a self-healing function.</p> <p>Test Case:</p> <ol style="list-style-type: none"> 1. Log into the VM called 'demo-vm1' via the console of the Openstack Horizon GUI (use contrail/contrail123 credentials) 2. Run command to create a traffic flow 	Current
----	--	--	---------

		<ul style="list-style-type: none"> • <code>iperf -c <ip address of demo-vm2> -b 500m -l 32k -w 128k -i 10 -t 10000</code> Note: the IP address of demo-vm2 can be found either from the Openstack Instances GUI or the AppFormix Dashboard for the vm instance. <ol style="list-style-type: none"> 3. Log into the AppFormix Dashboard and navigate to the 'demo-vm2' instance 4. Monitor the traffic usage on the instance increase <ul style="list-style-type: none"> • View the ingress traffic load from an Instance perspective using the <code>instance.network.ingress.bit_rate</code> chart • Move to the host chart and view the ingress traffic load from the underlying host perspective using the <code>host.network.ingress.bit_rate</code> chart 5. Create an alarm condition with a static threshold value below the current traffic flow rate <ul style="list-style-type: none"> • Set the threshold to 400Mbps, the alarm should first go into a learning state • Check the alarm condition is raised after the defined learning period • Navigate directly to the instance chart from the alarm description and examine the alarm alert icon on the <code>instance.network.ingress.bit_rate</code> chart x-axis Note: Check that alert indicators have appeared on the chart x-axis. Click on the arrow icon and inspect the alarm condition. Alerts are raised instantaneously as the condition is detected by the AppFormix agent. • Navigate to the host and check there is no alarm condition on the <code>host.network.ingress.bit_rate</code> chart x-axis 6. Kill the traffic flow and check the alarm condition clears 	
5b	Dynamically learnt threshold alarm generation on an infrastructure parameter	AppFormix automatically baselines for metric values over a defined period of time and automatically sets threshold values. Test Case:	Current

		<ol style="list-style-type: none"> 1. Create a dynamic alarm for analyzing a metric, for example instance.cpu.usage. Apply the rule at the Aggregate level for the 'appformix' project. 2. Consult external reference material on the merits of using k-means clustering and ewma statistical analysis modelling algorithms Use K-means for this rule. 3. Dynamic baselines may be created for different granularity i.e hourly, daily, weekly, monthly. Create and visualize the charts for the alarms based on some metrics Use hourly for the purposes of this sandbox environment. 4. Allow the baseline machine learning algorithm to run for some time. 5. Return to this test case later on and create a condition that significantly modifies the previous operating values of a defined metric. Check alarm conditions are raised. 	
5c	Composite Rule event generation	<p>AppFormix can compose different alarms and provide a holistic policy view from the perspective of several alarms.</p> <p>Test Case:</p> <ol style="list-style-type: none"> 1. Create a composite alarm based on several alarms rules 2. Apply, 'Any' based policy on composite alarms 3. Define a weighted policy on composite alarms 	Current

6. Infrastructure Topology Visualization, Resource Consumption and Capacity Reporting

Business value: Higher degrees of topology visualization leads to operation cost reduction of managing complex network environments.

Differentiation: Navigation of both underlay and overlay network layers together with infrastructure wide consumption Heat Maps.

	Title	Description	Availability

6a	Heat Map of consumed resources	<p>AppFormix provides a visual depiction of the relationship between physical host and virtual machines executing on that host. It also provides a representation of the infrastructure performance and consumption.</p> <p>Test Cases:</p> <ol style="list-style-type: none"> 1. Navigate to the Heat Map tab and assess the GUI and filter behavior in the context of the different resource menu categories selected (i.e. Infrastructure, Aggregates, Hosts, Projects, etc.) 2. Place the mouse over each resource icon to examine the detail 3. Run a resource consumption test. One of the test from Section 3 can be used to vary the resource consumption for a given metric 4. Select appropriate filters in Heat Map and ensure the resource icons are correctly labelled and colored according to the expected temperature scale of the resource. 5. Look for any risk and health conditions across the Infrastructure, Aggregates and the AppFormix project. <p>Colour code: GREEN < x% (eg - 70% of baseline) YELLOW between x – y% (ex - 70-85%) RED otherwise</p>	
6b	Overlay topology visualization	<p>Overlay networks are automatically learnt if a Software Defined Network controller exists (e.g. Contrail Networking). Alternatively, the topology of network resources can be imported into the system.</p> <p>Test Case:</p> <ol style="list-style-type: none"> 1. Navigate to the AppFormix Network Topology tab 2. Expand the nodes to see the topology linkages and values 	<p>Current</p> <p>Overlay topology is discovered from Contrail in the /TRY sandbox</p>
6c	Network topology Heat Maps	<p>Heat Maps can be used to query network elements for metrics against a baselined environment. These maps are available for Network Elements hosts and instances in the network</p>	<p>Current</p>

		<p>Test Case:</p> <ol style="list-style-type: none"> 1. Log into OpenStack Horizon and re-run the traffic test on 'demo-vm1' using: iperf -c <ip address of demo-vm2> -b 500m -l 32k -w 128k -i 10 -t 10000 2. In the Network Topology Overlay view, set the filter to Type: instance and Metric: instance.network.egress.bit_rate The 'demo-vm1-vnet' and demo-vm2-vnet' should be outlined in red indicating a high usage 3. Click on the hosts icon for each of the virtual networks and then expand the instances on the nugget-compute host. 'demo-vm1' should be red in colour and 'demo-vm2' should be green. This is because the traffic is being generated in 'demo-vm1' and egressing to 'demo-vm2' 4. Click on the outer box icon of 'demo-vn1-vnet' and view the 'in' and 'out' statistics. Repeat for 'demo-vn1-vnet'. 5. Modify the filter for Metric to instance.network.ingress.bit_rate and the colour of the VMs should change. This is because the traffic is entering (ingress) into 'demo-vm2'. 6. Change the Type filter to host. The attachment of each VM on the host to the vnet should be colour coded. 7. Click on one of the VM instances attached to the nugget-compute node and click on the expanded name icon. The screen should navigate to the dashboard view for the specific VM. 	
6d	Capacity planning tools	<p>AppFormix supports capacity planning tooling to help analyze the current and historical usage of the cloud resources. This can be used for trend analysis and predicting when new cloud infrastructure builds are required.</p> <p>Test Case:</p> <ol style="list-style-type: none"> 1. Navigate to the AppFormix Plan tab and modify the context of the navigation menu bar. The context of the reporting in the main window adjusts based on the infrastructure component context. 	Current

		2. Examine the details of the Overview, Query and Charts tabs	
6e	Infrastructure consumption and reservation reporting	<p>AppFormix reports can be used to determine the current and historical consumption values for the cloud estate.</p> <p>Test Case:</p> <ol style="list-style-type: none"> 1. Run a report on Project and Host 2. Examine the output of the reports and look for any flavor resizing recommendations 3. Examine the custom reporting tab option 	

7. Cloud Resource Consumption Chargeback and Billing

Business Value: Chargeback and billing functions enable cloud infrastructure divisions to recuperate costs from the main consumers of the infrastructure. This capability can be used to recover costs and improve the top line of the business.

Differentiation: Highly granular billing metrics can be utilized to improve business models that were previously out of reach.

	Title	Description	Availability
7a	Rate card cross-chargeable and billable metrics	<p>AppFormix supports a rate card that enables the construction of billable metrics across the complete cloud infrastructure.</p> <p>Test Case:</p> <ol style="list-style-type: none"> 1. Define rate card costs for a set of infrastructure metrics 2. Set up defined organisations based on project usage <p>Note the Chargeback tab will not become active until the next billing report run is scheduled. The current sandbox will expire before the first monthly report becomes active. Refer to AppFormix documentation on the format of the Chargeback report.</p>	Current

AppFormix Demo Use Cases Available from Alternative Juniper POC Labs:

Many complex AppFormix automation demos are available but require a more comprehensive lab configuration than is provided by the current /TRY sandbox environment. Please speak to your Juniper account representative for scheduled demos of additional AppFormix capabilities.

<p>3. Analysis of Operational State of Infrastructure Resources Business Value: Operation cost reduction of navigating the topology of managed infrastructure resources. Differentiation: Complete visibility for resource metric objects with relationships to physical and virtual infrastructure resource consumption. Enables simple investigations into ‘noisy workloads’ and the impact on infrastructure consumption.</p>			
3d	Enable software process set monitoring	<p>AppFormix monitors the health of software processes on physical infrastructure hosts and virtual instances where a distributed AppFormix Agent is installed. AppFormix APIs and its ansible playbooks together program a process-set on a host or a tag on hosts for monitoring.</p> <p>Test Case:</p> <ol style="list-style-type: none"> 1. Visualize metrics on hosts or aggregates where plugin for Process monitoring is enabled 2. Configure alarms on the metrics being stream for particular processes 3. Look for any risk and Bad Health status icons with these metrics. 	Current

<p>4. Closed-loop Enforcement of Resource Management SLAs Business Value: Higher degrees of customer SLA leading to incremental service revenue options. Operation cost reduction of managing infrastructure resources by providing near real-time notification of infrastructure health issues. Differentiation: Real-time state-based compliance events can be used to notify operations staff or send to self-healing automated functions</p>			
	Title	Description	Availability
4d	Alarm notification sent to external SaaS incident response platform	<p>AppFormix detects resource exhaustion and sends a notification to an external SaaS platform</p> <p>Test Case:</p>	Current

		<ol style="list-style-type: none"> 3. Configure the Notification Settings under the AppFormix Settings pane so that notifications are send to an external SaaS system 4. Run a burn script on a physical host 5. Set an alarm conditions to detect the CPU load threshold crossing and send a notification 6. Observe the notification arrive at the external system 	
4e	Closed-loop instance workload migration	<p>AppFormix detects resource exhaustion and sends a workload migration request to auto-heal the service.</p> <p>Test Case:</p> <ol style="list-style-type: none"> 1. Install the Openstack workload migration listener module 2. Run a burn script on a physical host 3. Set event conditions to activate migration events 4. Observe VM workload migrations onto another compute host 	Current
4f	Closed-loop Tag group automations	<p>AppFormix sets state monitoring for a Tag. Whenever the state or SLA for the Tag or object gets violated an event can be triggered.</p> <p>Test Case:</p> <ol style="list-style-type: none"> 1. Configure Alarms for some Device metrics 2. When Alarms gets violated for the Device send a notification Json to the listener. <p>Take Action at the listener for the notification example Apply firewall policer rule on ingress-bandwidth for a host.</p>	Current

8. Network Device Visualization and Self-healing Automation

Business value: Higher degrees of network visualization and automaton leads to operation cost reduction of managing complex network environments.

Differentiation: Self-healing networks built on the foundation of AppFormix analytics and automations.

	Title	Description	Availability
8a	Monitoring health of SNMP collector to each managed device as well as metrics for different Network elements exposing SNMP metrics	<p>AppFormix collects metrics from SNMP MIBs that are supported by most multi-vendor network devices.</p> <p>Test Case:</p> <ol style="list-style-type: none"> 1. Navigate the Dashboard View to the Network Devices pane and examine the source collector for each device 2. Identify a device with a SNMP source 3. Click on the reports icon and view metrics value obtained from the device 4. Expand and shrink the time window to examine the historical trend line of metrics 5. Configure alarms on metrics 6. Configure which MIBs to monitor for any given device 7. Add MIBs providing output specified by a particular format for monitoring 8. If collector for an SNMP device dies. Fail-over collection to another collector. API expresses collector state of a device. 9. Generate reports for metrics Device and Device Interfaces 10. Distribute collection to horizontally scaled tier of Agents 	Current
8b	Monitoring health of JTI collector to each managed device as well as metrics for different Network elements exposing JTI streams	<p>AppFormix collects receives UDP stream of Junos Telemetry Interface (JTI) feeds that are supported by Juniper network devices.</p> <p>Test Case:</p> <ol style="list-style-type: none"> 1. Navigate the Dashboard View to the Network Devices pane and examine the source collector for each device. 2. Identify a device with a JTI source. Check there are no heartbeat risk or health warnings 3. Click on the reports icon and view metrics value obtained from the device 	Current

		<ol style="list-style-type: none"> 4. Expand and shrink the time window to examine the historical trend line of metrics 5. Configure alarms on metrics 6. Configure which Sensors to monitor for any given device 7. Add Sensors providing output specified by a particular format for monitoring 8. If a device does not report within a ConfigurableConstant * ReportingInterval report a device.heartbeat metric 9. Distribute collection to horizontally scaled tier of Agents 	
8c	Dynamic Alarms per Network Element interface	<p>Provide alarms baseline per interface for a Network Device</p> <p>Test Case:</p> <ol style="list-style-type: none"> 1. Add Alarms for 1 particular interface on a Device 2. Add alarms for 1 particular interface on an Aggregate of Devices 3. Add alarms for multiple common interfaces for aggregate of Devices 4. Visualize dynamic baseline as an aggregate or per-interface for a device. 	Current
8d	Network self-healing with Northstar integration	<p>AppFormix and Northstar can be integrated together so that AppFormix informs Northstar of network health issues. Northstar can be used to self-heal the network services and re-routing traffic away from problem areas.</p> <p>Test Case:</p> <ol style="list-style-type: none"> 1. Configure AppFormix and Northstar intergration 2. Create a bad health condition that AppFormix can identify using SNMP or JTI 3. Confirm Northstar puts impacted devices into Maintenance Mode and traffic is routed away from problem areas. 	Current

9. Public Cloud Infrastructure Visibility

Business value: Visibility of public cloud infrastructure elements for high precision and customizable metrics. Cost saving based on right-sizing flavor recommendations to the user.

Differentiation: Single pane of glass for on-prem and off-prem resources.

9a	Cloud Visibility	<p>AppFormix provides an adapter based approach to integrate with Cloud Platform APIs. Support Cloud platforms out of box are AWS, GCP, Azure. We can pick one platform for the purposes of the POC</p> <p>Test Case:</p> <ol style="list-style-type: none">1. Enable adapter based monitoring for cloud-watch metrics per the credentials of an admin2. Enable agent based monitoring where metrics with a higher precision maybe needed3. Visualize metrics on dashboard and create Alarms for entities.4. Enable metrics for plugins5. Enable metrics for processes running for an agent.	Current
9b	Cloud Right Sizing Reports	<p>AppFormix provides report generation and recommendation on flavor sizing based on dynamic usage.</p> <p>Test Case:</p> <ol style="list-style-type: none">1. Create a report on cloud instances2. Visualize sizing recommendation based on workload and resource consumption for the cloud instance.3. Apply tags on cloud instances and visualize key metrics on reporting based on tags.	Current
9c	Cloud Network Elements	<p>Cloud based network elements can be added into AppFormix using network telemetry. Examples are cloud virtual network elements like vSRX.</p>	Current

Document History:

Revision	Date	Modification Description	Author
1.0	16 th April 2018	First document build	Scott Alexander
1.1	17 th April 2018	Extended section for further tests in Juniper POC labs	Scott Alexander
1.5	14 th May 2018	Addressed test case modifications	Scott Alexander