Aviz Networks

WHITEPAPER

UNLOCKING ENHANCED VISIBILITY WITH OPEN-SOURCE NETWORK PACKET BROKER SOLUTIONS

Madhu Paluru (Director, Engineering at Aviz Networks)

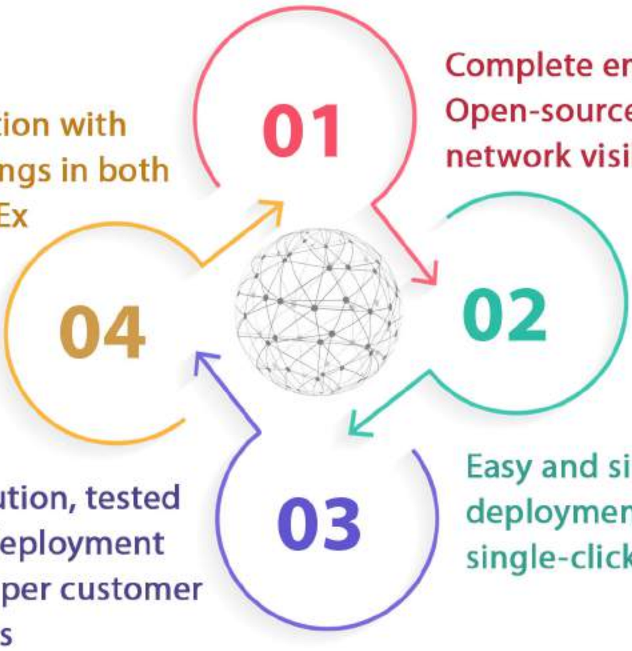
Lakshmi Krishnaswamy (University of California, Santa Cruz)

Overview

Network packet capture and analysis is very crucial for any enterprise network. The use cases of network visibility range from inferring insights about the kind of traffic entering to the network to intrusion detection and understanding internal load-sharing behavior and performance. Each company usually has its own enterprise network solution it employs to leverage visibility into its networks. Existing solutions like cPacket, Gigamon, etc. are much sought out and used. However, they come with the drawbacks of high CapEx and OpEx costs, scalability issues, and often limited support and assistance attributed to the nature of vendor lock-in.

To address the drawbacks and challenges of the existing commercial solutions, we present the "Cost-efficient Open Source Network Visibility Solution from Aviz Networks". The proposed solution harnesses cost-efficient open-source tools for comprehensive visibility and analysis. This framework is truly based on open networking principles and open-source technologies. These include open networking switch hardware running software-defined OPB NOS built on completely open-source NOS SONiC (Software for Open Networking in the Cloud) providing aggregation, filtering, and load balancing of monitored traffic. The monitoring solution is completely designed using open-source analytics software for packet processing, storage, and visibility on commodity x86 server nodes.

The key contributions of the paper can be summarized as:



01 Complete end-to-end Open-source solution for network visibility and analysis

02 Easy and simplified deployment with single-click installation

03 Scalable solution, tested for various deployment scenarios as per customer requirements

04 Low-cost solution with huge cost savings in both CapEx and OpEx

Requirements

Typically, traffic monitoring can be classified into various categories and the following data points play a critical role in it:

- ✓ Based on deployment zones
- ✓ Kinds of traffic
- ✓ Application use cases
  - Traffic monitoring from east-west, consisting of traffic within their data center
  - Traffic monitoring north-south, consisting of traffic entering a given data center network
- ✓ Deployment size and scale
- ✓ Placement of deployment
- ✓ Deployment model (centralized vs. distributed)
- ✓ Low CapEx and OpEx

For instance, let's take an example of an e-commerce company that consists of multiple data centers that can be geographically distributed.

The specific requirements of this use case are:

01 Monitoring the kind of traffic that enters these networks - **imperative for security inspection and analyzing traffic patterns.**

02 Having a centralized viewer that has easy access to all the different sites with ability to view dashboards, run queries, and perform filtering for different sites - **for scalability across different sites with a single pane of view**

03 Centralized management of network packet broker rules and filters – **to enable a single point of control for multiple packet broker switches**

04 Network planning with placement design for storage nodes that are scalable, handle huge amounts of data, and are resilient to failures - **for disaggregated deployment of storage databases across sites**

05 Delivering all of the above in a low-cost, optimized end-to-end solution package.

Competitive Analysis

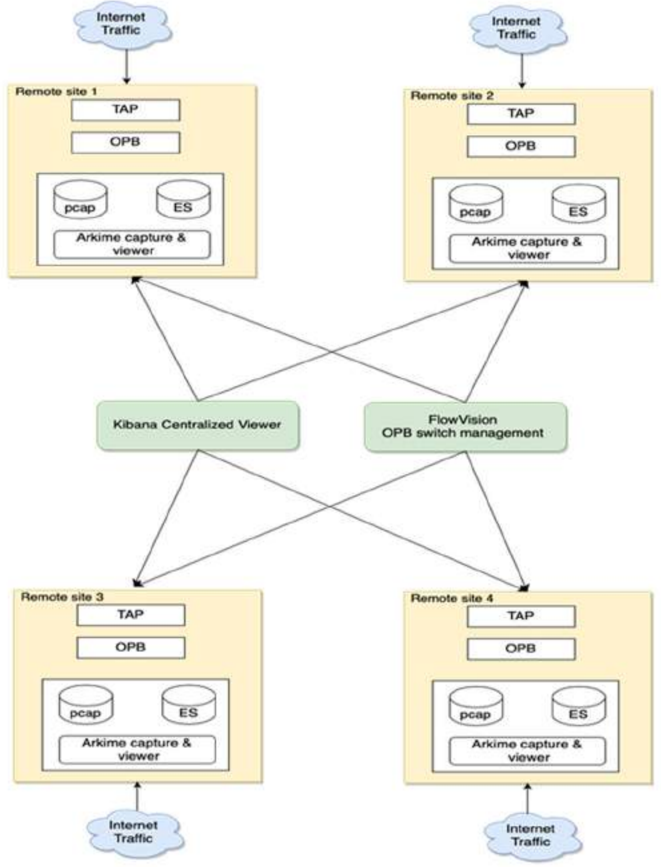
As per our research, there are a few solutions out there that try to address some of the customer requirements. However, they come up with their own pros and cons. In many instances, these are not offered as a one-stop solution that delivers network packet broker and capture and analysis functions. In fact, they have to be composed of different enterprise products to achieve an end-to-end solution.

We found cPacket and Gigamon are the most prominent solutions that help customers address the need to capture packets at the line rate and the storage requirements for PCAP (packet capture) and metadata. They offer mostly in-house products like proprietary network packet broker, storage, and visualization products. In a nutshell, there is no one-stop solution available and the customers end up buying multiple products to get a comprehensive solution. The main drawbacks of all these solutions are that they are locked into a single vendor and the customer ends up paying huge costs for deploying solutions and renewed annual technical support. Another drawback is its difficulty in adapting to changes. Changing or upgrading a packet broker or the visualization software is non-trivial and integrating a product from a different ecosystem would be expensive and time-consuming.

Aviz Networks offers a one-stop solution that satisfies all customer requirements such as line rate packet capture, metadata, and packet storage with a completely open-source based solution. The solution comprises all the open-source components such as an open-network packet broker built on top of open-source SONiC and packet capture along with vsibility solutions based on Arkemi, Elastic, and Kibana. The solution also provides various disaggregated deployment options for customer needs. Since it is completely built on top of open source, the customers never get locked into any single vendor and are not required to spend on annual support costs. This makes deployment and maintenance much easier and cost-efficient. It is also backed by great community support for any upgrades and changes in the pipeline. Overall, with an open-source based visibility solution, Aviz Networks' customers will enjoy up to 80% total cost ownership (TCO).

Solution Architecture

Given the problem statement on the customer requirements and limitations of current solutions, we present: "Aviz Open Packet Broker (OPB) Visibility Solution". The proposed solution addresses the limitations of the current solutions and fulfills customer requirements in an optimal manner. This complete end-to-end pipeline is composed of open-source software, right from the switch networking to capture, storage, and visualization tools.



Software Components

OPB NOS

Aviz Network's Open Packet Broker (OPB) can run on any white box switch agnostic to the underlying hardware. The proposed solution stack comprises OPB NOS as the network packet broker built on top of open-source SONiC which is the first point of traffic input in the pipeline. OPB NOS can be leveraged to accept and drop traffic based on custom rules such as creating a block list of IP addresses which protocols to accept and many more.

Some critical functionalities of OPB NOS include (but are not limited to this):

- Switch monitoring
- Packet aggregation
- Load balancing
- One-to-many traffic forwarding
- Packet filtering
- Packet splicing
- Loopback-mode port configuration
- Supports up to 400GB of line rate.

Arkime

Arkime is a large-scale, open source, full packet capture system. This component in the pipeline captures all the traffic coming in from the Aviz Networks' Open Packet Broker. Arkime offers a wide variety of functionalities from traffic filtering to saving PCAP captures, and searching for specific sessions based on IP addresses, port numbers, protocols, etc. The generated metadata is stored in an Elasticsearch backend in the form of JSON documents.

Arkime
Full Packet Capture

Arkime has a GUI that can be used to view different sessions and download session info in PCAP format, JSON format and also as a CSV file. In addition to capturing live traffic, Arkime also supports the import of PCAP traffic capture to perform further analysis.

Different placements of Arkime have been tested and explored to address customer requirements that involve multiple distributed traffic sites. The ideal setup is to have every distributed site get a local Arkime instance that captures live production traffic and writes to its own local ElasticSearch instance.

ElasticSearch

ElasticSearch is the backend database that is used to store all the metadata generated by Arkime. It is an open-source distributed, RESTful search and analytics engine capable of addressing a growing number of use cases[3]. Raw data flows into Elasticsearch from a variety of sources; in this case, the captured data is from Arkime. Data ingestion is the process by which this raw data is parsed, normalized, and enriched before it is indexed in Elasticsearch. Once indexed in Elasticsearch, users can run complex queries against their data and use aggregations to retrieve complex summaries of their data. This data can be accessed from Kibana to create powerful visualizations. Elasticsearch is fast, distributed by nature, and comes with a wide set of features.

As seen under Arkime, in order to best support scalability and storage requirements, each distributed traffic site has its own ElasticSearch instance to store all the metadata.

Kibana

Kibana is an open-source software that is a powerful visualization and data exploration tool. In the solution pipeline, Kibana is the centralized viewer that comes with predefined dashboards highlighting different use cases to infer helpful insights.

In addition to these predefined dashboards, Kibana also has a strong search analytics interface. Using the "discover mode" in Kibana, it is possible to run queries based on the (KQL) "Kibana Query Language". One can also filter based on protocols, timestamps, and even hostname to differentiate which traffic came from which Arkime capture node. Kibana also acts as the user interface for monitoring, managing, and securing an Elasticsearch cluster.

Some of the advantages of the dashboards can be summarized as follows:

- Classifying the different types of application traffic
- Understanding load distribution across servers
- Aggregation packet metadata over time with different source agents sending requests to various destination hosts
- Understanding Top Talkers"
- Detects denial of service attacks.
- Protocol breakdown.

FlowVision

FlowVision is yet another tool from Aviz Networks that is a centralized GUI to manage all the different OPB switches and can be deployed in any architecture. It provides an easy platform for network administrators to add switches that need to be configured and monitored. It also enables the configuration of the rules and interfaces and monitors health and syslog reports.

Solution Deployment Models

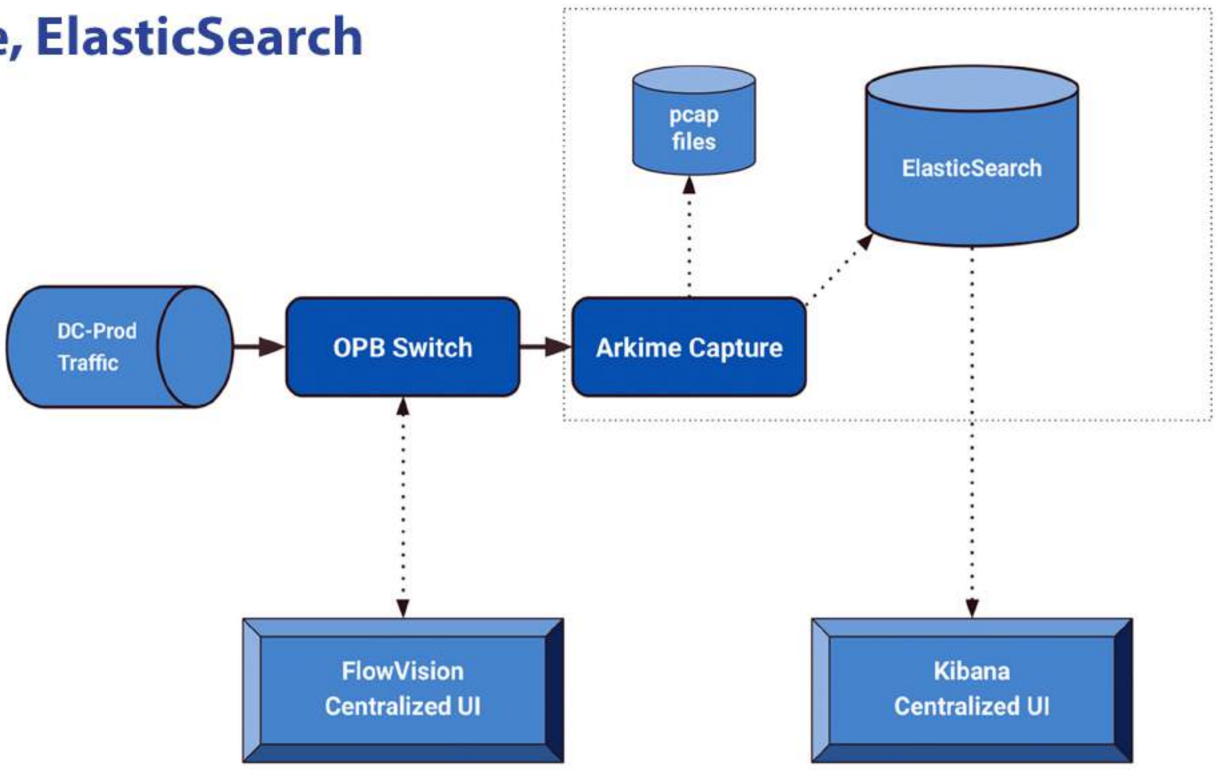
Centralized Kibana + Distributed Arkime, ElasticSearch

In this described architecture, live data center production traffic is captured by the OPB switch—filtering for any traffic, and applying block list rules if any. This traffic is captured by Arkime on the listening interface, and Arkime generates Metadata and sends it to a local ElasticSearch database.

PCAP format of the captured traffic is also saved by Arkime. Every data center site that is geographically distributed runs an Arkime and ElasticSearch instance locally. In order to search across geographically distributed sites and display results centrally - solution enables "cross-cluster search" through Kibana. Cross-cluster search is designed to be scalable across geographically distributed sites. By setting up a local ElasticSearch (ES) instance for Kibana, it queries all the remote ElasticSearch instances and displays a centralized view of all the

The main advantage of this deployment is scalability. By disaggregating the Elasticsearch instances, users can scale storage horizontally with incoming traffic per deployed site. Another critical advantage is data sovereignty and customer data will be stored in the site.

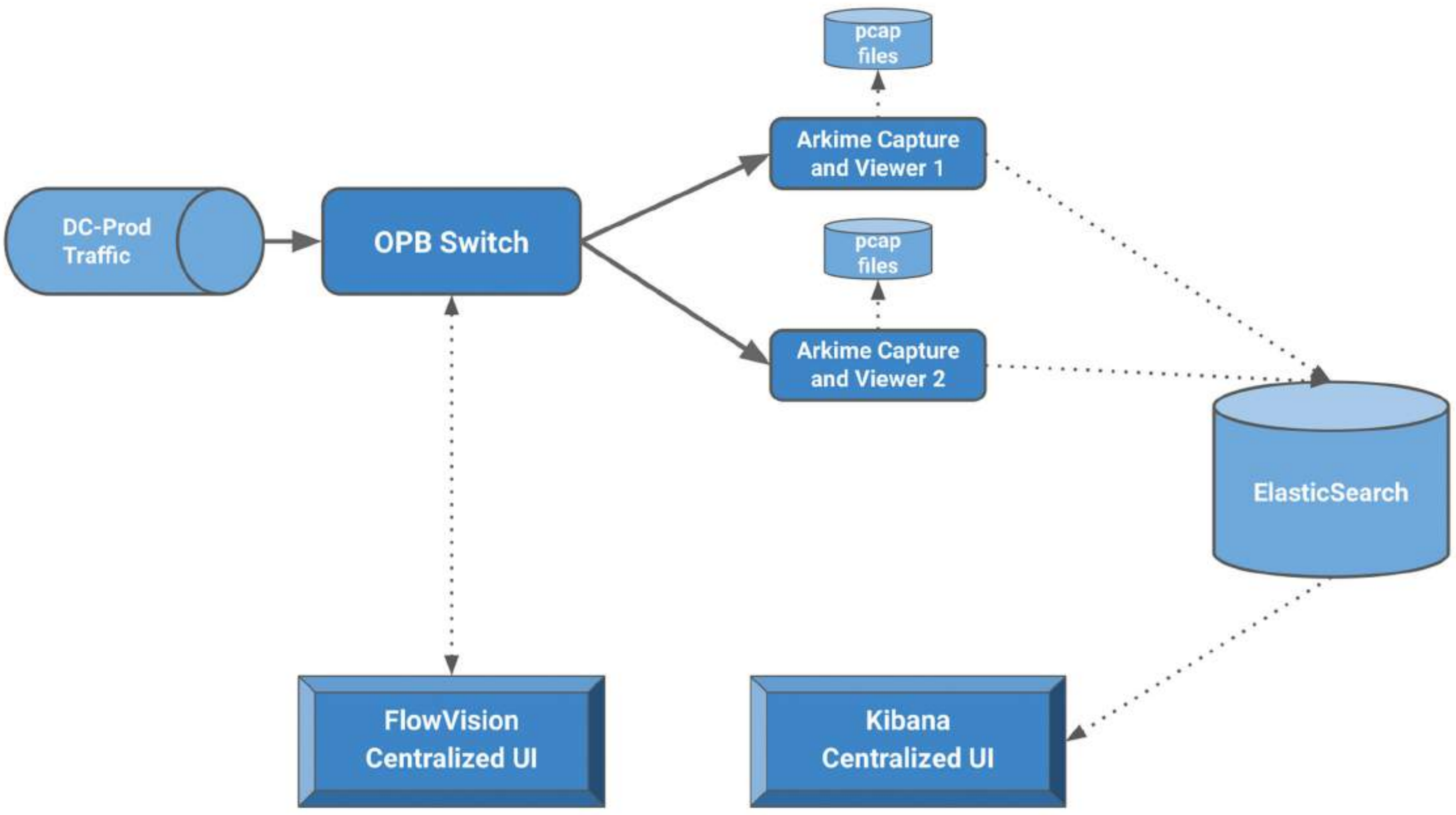
Multi site deployment with Centralized Kibana + Distributed Arkime, ElasticSearch



Centralized Kibana + ElasticSearch, with Distributed Arkime Capture Nodes

In this described architecture, the major difference between the setup described above is to have a centralized ElasticSearch instance that all Arkime nodes write to. While this might not be the best-suited model for a geographically distributed deployment, this can be used for monitoring a local data center network or a single office's internal network.

Multi Site Deployment with Centralized Kibana, Elastic with distributed Arkime


pcap files
Arkime Capture and Viewer 1
DC-Prod Traffic
OPB Switch
pcap files
Arkime Capture and Viewer 2
ElasticSearch
FlowVision Centralized UI
Kibana Centralized UI

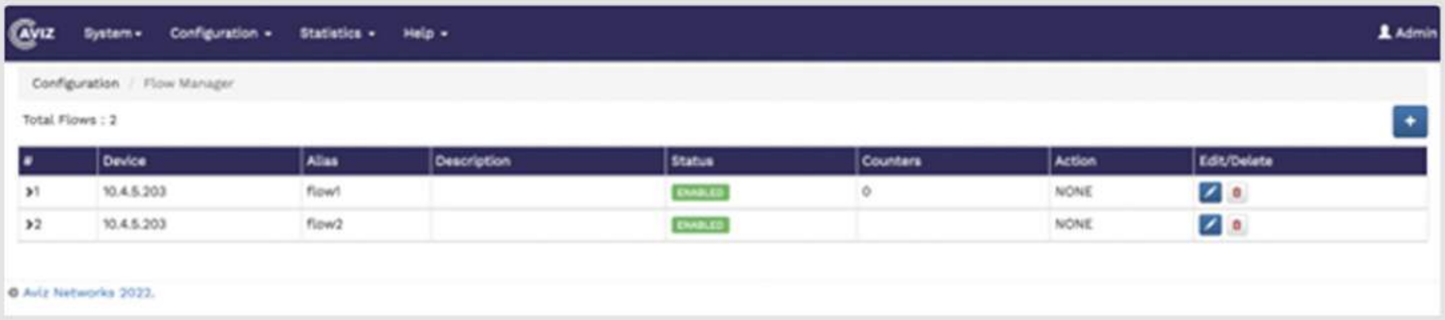
Single-Click Deployment

In order to deploy this solution framework, we provide a single click installation script to deploy the following BOM (Bill of materials).

Setting up OPB NOS

The first step of the solution is to set up the rules in the switches running OPB NOS under various roles. Below example provides a simple CLI example for setting up an aggregator and load balancer for filtering and forwarding IP traffic along with truncation of packets beyond 64 bytes. The same can also be set up using the FlowVision GUI.

OPB FlowVision GUI



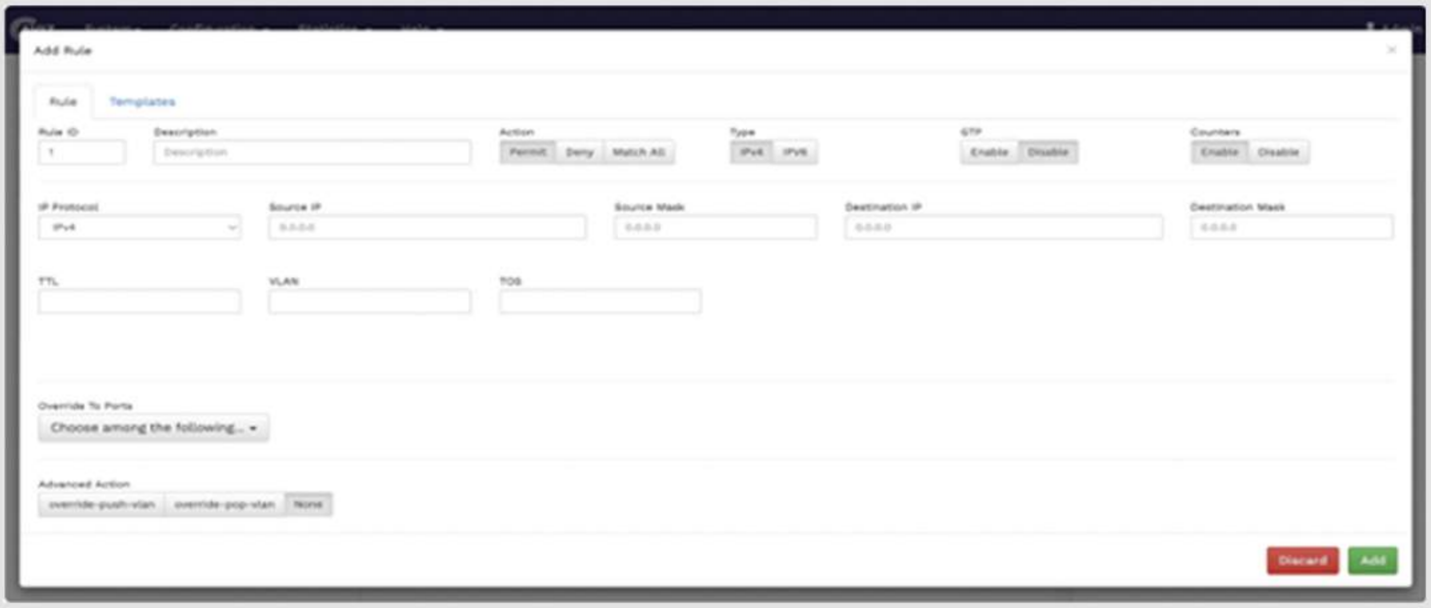
Configuration / Flow Manager

Total Flows : 2

#	Device	Alias	Description	Status	Counters	Action	Edit/Delete
1	10.4.5.203	flow1		ENABLED	0	NONE	
2	10.4.5.203	flow2		ENABLED		NONE	

© Aviz Networks 2022.

OPB FlowVision Rule Manager



Rule ID: 1
Description: Description
Action: [✓ Permit] Deny Match All
Type: [✓ IPv4] IPV6
GTP: Enable [✓ Disable]
Counters: [✓ Enable] Disable
IP Protocol: IPv4
Source IP: 0.0.0.0
Source Mask: 0.0.0.0
Destination IP: 0.0.0.0
Destination Mask: 0.0.0.0
TTL:
VLAN:
TOS:
Override To Ports: Choose among the following...
Advanced Action: override-push-vlan override-pop-vlan [✓ None]

Setting up Arkime Capture and Viewer and ElasticSearch Storage

Aviz Networks provides a simplified tool that can be used to inflate or deflate entire infrastructure on a single or multiple servers. Once Arkime is up and running, it will start capturing packets from configured network interfaces which are connected to OPBNOS running network aggregators and load balancers.

In order to set up ElasticSearch and Arkime, we run the ArkimeESsetup.sh script from the same folder we have OPBAnalyzer.tar.gz in. This single script takes care of setting up both ElasticSearch and Arkime on the same site to support the scalable, disaggregated model discussed above. It takes the following user inputs, in turn making it easy to deploy and customize for every requirement.

```
OPB_Analyzer$ sudo ./ArkimeESsetup.sh
Extracting OPB_Analyzer.tar.gz using 'tar -zxvf'...
OPB_Analyzer/
OPB_Analyzer/stopElastic.sh
OPB_Analyzer/arkime.tar
OPB_Analyzer/stopArkime.sh
OPB_Analyzer/ArkimeUploadHelper.py
OPB_Analyzer/ArkimeHelper.sh
OPB_Analyzer/startElastic.sh
OPB_Analyzer/elasticsearch.tar
OPB_Analyzer/ipv4-address-space.csv
OPB_Analyzer/.DS_Store
OPB_Analyzer/arkime_update_geo.sh
OPB_Analyzer/manuf
OPB_Analyzer/startArkime.sh
Running startElastic.sh
OPB_Analyzer$ Enter the node-name to be assigned for ES instance (e.g., es01) : es01
OPB_Analyzer$ Enter host:port to set as discovery.seed_hosts (e.g.,10.x.x.x:9300) 10.x.x.x:9300
OPB_Analyzer$ Enter the absolute path where ES data needs to be stored (e.g., /data) :
/home/admin/data
Loading docker images...
Loaded image: docker.elastic.co/elasticsearch/elasticsearch:7.17.3
vm.max_map_count = 262144
WARNING: Published ports are discarded when using host network mode
e98f38a6ccef8e8a9dd4c796df309b5eb2c67b943f440a43586b29297ceaec39
waiting for initialization...
```

```
Running startArkime.sh
OPB_Analyzer$ Enter Semicolon ";" separated list of interfaces to listen for live traffic: ens64
OPB_Analyzer$ Enter the IP address of the ES instance this Arkime instance connects to : 10
OPB_Analyzer$ Enter the hostname for this Arkime instance  : site1
OPB_Analyzer$ Enter the admin-password for this Arkime-UI : admin
OPB_Analyzer$ Enter mode for Arkime capture (on/off) (e.g.,on) : on
OPB_Analyzer$ Enter mode for Arkime viewer (on/off) (e.g.,on) : on
OPB_Analyzer$ Enter the absolute path where Arkime data needs to be stored <(e.g.,/data) :
/home/admin/data
Loading docker images...
Loaded image: avizdock/docker-arkime:latest
vm.max_map_count = 262144
2151132d9ac9b8bac0a01ffde9676153860f5c14004dc333b2e4c3c60b7855b0
Access Arkime Viewer at:
http://<host>:8005
username: admin

waiting for initialization...
Running ArkimeHelper.sh
Running ArkimeHelper.sh
                    Successfully copied 3.07kB to arkime:/opt/arkime/bin/arkime_update_geo.sh
                    Successfully copied 2.16MB to arkime:/opt/arkime/bin
                    Successfully copied 25.1kB to arkime:/opt/arkime/bin

Running ArkimeUploadHelper.py : pre-req : python3
                     Successfully copied 17.9kB to /home/lakshmi/disaggregatedset
                    up/test_scripts_v4/OPB_Arkime_setup/OPB_Analyzer/config.ini
                     Successfully copied 17.9kB to arkime:/opt/arkime/etc/config.ini

Restarting Arkime container
arkime
```

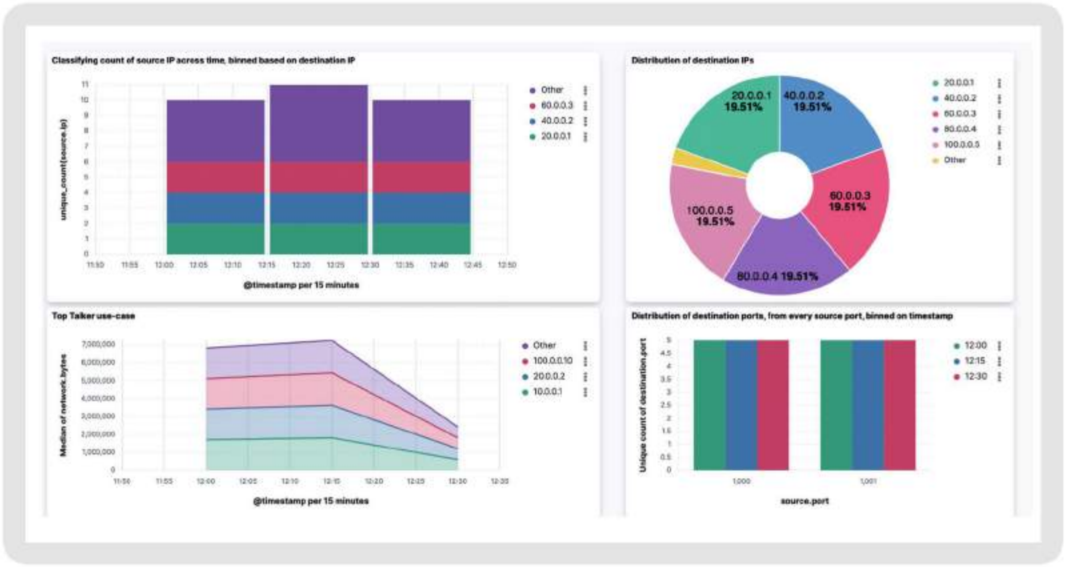
Setting up Kibana Visualization

In order to set up Kibana, we run the ./Kibanasetup.sh in the same folder we have the OPB_Kibana.tar.gz

This will be set up only in the single server to serve as the centralized single pane of view.
➡ **Enter the IP address of the local ES instance Kibana should connect to eg:10.x.x.x**

Once Kibana is up and running. Login to the Kibana UI at http://[server-ip]:[5601] to create a dashboard from Arkime data sessions. Next, navigate to index patterns and create an index pattern for Arkime data sessions from available fields drag-&-drop that need to be monitored and create a visualizer to add to the dashboard.

Centralized Kibana Dashboard



Steps to configure remote cluster through Kibana UI

1. In Kibana UI, need to add a remote cluster, Kibana -> Stack Management -> Remote Clusters
2. Restart Kibana
3. Add both the index patterns: arkime_-session* and es2-docker-cluster:arkime_-session
4. (Replace es2 with your specific node_name, and same for IP address for discovery.seed_hosts)

CapEx and OpEx

One of the leading network visibility solutions quoted $30K for hardware and software appliance and a 20% maintenance fee, which then comes up with an additional annual support cost of ~$6,000.

Components (per Site)	Proprietary	Open-Source
Hardware and software appliance	~$30k	~6K (hardware Only)
		$0 (software)
Annual software support	~$6k	$0

Conclusion

In this work, we presented the "Aviz Open Source Based Visibility Solution", which harnesses cost efficiency with truly open source tools for comprehensive visibility and analysis. The proposed solution completely leverages the open source tools in every stage of the pipeline, thereby delivering on reduced cost. It also opens customers to the world of open source with a large community to provide support and guidance at all times. The discussed architecture with disaggregated Arkime + ElasticSearch instances, with cross-cluster searching enabled is scalable and designed for geo-distributed sites.

Aviz delivers switch platform agnostic, easy-to-use applications for network orchestration, visibility, and assurance. Aviz also provides SLA-backed disaggregated support for multi-vendor SONiC deployments in data center and edge networks.

Address | Aviz Network Inc
US Headquarters
2150 N First St, Suite 442
San Jose, CA 95131

Contact | hello@aviznetworks.com
sales@aviznetworks.com