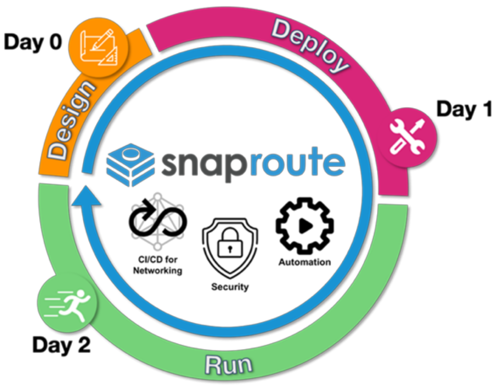
CN-NOS Release 1.0



Key Features

- CI/CD for Networking: CN-NOS is the industry's first Cloud Native Network Operating System to offer Continuous Integration/Continuous deployment. Since all the key protocols and services are containerized (as microservices) and orchestrated via an embedded version of Kubernetes, network teams can synchronize themselves with compute and storage.

- By natively leveraging Kubernetes, it's now possible to load only what is needed depending on the use case/deployment scenarios: gone are the days when you load an entire protocol stack even if you need only a fraction of it.

- Another Industry first: upgrading the key components (protocols/services) can be done in a "surgical" manner without the fear of disrupting the entire system or the need of rebooting the entire switch. This allows a massive reduction in time and resources required to manage your data center networking fabric.
- Security compliance – now a reality: most organizations take significant risks by delaying security patches. With the ability to surgically patch/upgrade services/protocols, network operators can confidently apply these security patches in a timely manner.

- Highest level of automation/orchestration: by standardizing on the "de-facto" Data Center automation/orchestration platform – Kubernetes – operators can for the first time manage the data center from an end to end perspective without the need for intermediate layers (adding risks and complexity) delivering the highest level of predictability.

- Ultimately delivering "Day 2" operations: designing data center leveraging Hyperscalers guidelines has never been so easy. SnapRoute's CN-NOS provides the ability to manage the life cycle after the initial installation/provisioning in complete coordination with the rest of the DevOps teams – without any constraint.

Product Overview

The Cloud Native – Network OS from SnapRoute is the only fully containerized microservices NOS for disaggregated switches.

CN-NOS is the embodiment of a Network OS built from the ground up for operators, by operators. Embedded in its core functionality is the wisdom and battle scars of network engineers who helped design and build some of the world's largest and most scalable data centers.

With the rise of public cloud platforms, organizations are adopting management principles and toolchains to manage on-premise deployments in a similar fashion. The only thing left behind is the network. The only way to remove the network from its prescribed silo is to elevate it to the same level as the rest of the environment.

The network needs to be managed using the same Cloud Native tools that are being used to deploy containerized infrastructure. The network has to be built using an OS that adopts these principles and presents the network using models and paradigms that are native to the operators of Cloud Native environments. This solution cannot use add-ons, bolt-ons, extensions, or other proprietary and bespoke solutions – the network needs to be managed using Cloud Native tools, natively.

CN-NOS is packaged and delivered as a complete Network OS that installs via the ONIE bootstrapping environment onto disaggregated "whitebox" switches. It provides the full control-plane and data-plane functionality required for autonomous device operation – no fabric-level controller or proprietary management platform is required. Every CN-NOS installation includes full native Kubernetes support for direct device management via kubectl and kube-apiserver. Knowledge of Kubernetes is not a prerequisite for leveraging the benefits of the microservices-based platform of CN-NOS – a full-featured industry-standard CLI also ships with CN-NO

Microservices based NOS

- Industry First containerized NOS
- Isolates into *containers* the critical protocols and services to facilitate “Day 2” operations.
- Containers are automated/orchestrated (locally) by Kubernetes (“*K8s*”)
- Full Network Operating System: via a single image or a local repository for facilitating future patches/upgrades (similar to what operators do for the compute with a local Docker registry), CN-NOS is a complete Network Operating System including the base OS (Yocto) as well as all the protocols (L2, L3, L4) and services (HW Programing, Manageability, Serviceability ..) to deliver use cases in the context of the data center.

Open Networking / Open Source

- OCP/ONIE: as technologists across industries participate in the OCP community, it’s now possible to create more designs, making it possible for more companies to transition from their old, proprietary solutions to OCP hardware. As applications require more and more services, data and connectivity, this has to be done the most efficient, economical and sustainable way. Hardware must become a commoditized and evolving set of products optimized for these challenges. At SnapRoute, we believe that supporting OCP specifications (and specifically ONIE as networking related) is the way to go (see Hardware Compatibility List – HCL – at the end of this document).

- Built by one of the top 7 hyperscalers, Kubernetes - now hosted by CNCF (Cloud Native Computing Foundation) - brings the SW development and operations together by design. By using declarative, infrastructure-agnostic constructs to describe how applications (and in the case of a NOS protocols and services) are composed and how they interact, Kubernetes enables an order of magnitude increase in operability of modern SW solutions. SnapRoute did bring this environment into an embedded system with a limited set of resources (compared to a traditional server), optimizing how these resources are used and consumed. Some of these optimizations will be up streamed benefit the community (SnapRoute is a member of CNCF).

- Optimized for Modern Data Centers

- Hperscalers have been demonstrating how to deploy massively scalable data centers – leveraging modern principles - Leaf/Spine, Layer 3 (and in particular BGP combined to ECMP and BFD), very high degree of automation/orchestration. The first release of CN-NOS follows these guidelines to offera broader set of customers the same principles with the power of our microservices architecture focusing at solving Day 2 operations.

- High Performance Layer 3 Capabilities

- Leveraging the MetaSwitch stack, CN-NOS offers day 1 the best combination of features/functionalities, performance and interoperability, particularly in the context of BGP (eBGP/iBGP) as the technology to build modern data center fabrics. In addition, BGP was further optimized by *containerizing* the key functions (control, peering) which are executed in a multi-threaded manner. Associated to Equal Cost Multi Pathing (ECMP) and Bi-Directional Forwarding Detection (BFD), for both IPv4 and IPv6, CN-NOS a very high degree of redundancy.

- IPv4/IPv6
- CN-NOS offers a dual stack IPv4/IPv6 – BGPv4/MP-BGP, ECMPv4/v6, BFDv4/v6– a no compromise approach for those looking at the benefits of IPv6 in the context of the data center.
- Automation/Orchestration
- With a native integration of Kubernetes natively in the platform, CN-NOS offers a completely new paradigm to manage your switching infrastructure. Besides the fact that a specific list of protocols and services can be chosen for a given device/deployment, Kubernetes provides an advanced scheduler with self-healing capabilities (re-starting containers is done automatically) with automated rollouts and rollbacks, drastically improving predictability of a system during changes.

Security

- By choosing a highly differentiated architecture where all key protocols and services are containerized (*microservices*), the concept of fault domain becomes much smaller compared to a traditional/monolithic NOS.
- By default, process access leverages the principle of “least privilege” with “*namespaces*” providing process isolation.
- As each individual protocol and service can be patched/upgraded individually, network operators can be security compliant at a pace never executed before.
- CN-NOS leverages SELinux allowing users and administrators more control over access, in a highly granular manner (Role Base Access Management)
- All unsecure services are disabled by default – allowing only SSHv2/TLS/SSL based services/protocols

Troubleshooting

- “Watches”
- Natively integrated into Kubernetes, “watches” allow a client to fetch the current state and then watch for changes of any resource without missing any single update. Tracking how an interface comes alive with the suite of events from the physical layer (port up) to the highest layer (BGP sessions coming up step by step) has never been so easy.
- Ingress and egress port monitoring (“SPAN”) enable network debugging

LAYER 3	**BGP** – RFC 1163: Border Gateway Protocol – RFC 4271: Border Gateway Protocol v4 – BGP 1997: BGP Community Attributes – BGP 4360: BGP Extended Communities Attribute – RFC 4760: Multiprotocol Extensions for BGP-4 – RFC 7938: Use of BGP for Routing in large-scale Data Centers **IP** – RFC 768: UDP – RFC 783: TFTP Protocol – RFC 791: IP – RFC 792: ICMP – RFC 793: TCP – RFC 854/856/1091: Telnet – RFC 950: Internet Standard Subnetting Procedure – RFC 1027: Proxy-ARP – RFC 1531/1533/1534/1541/2131: DHCP – RFC 1542: Clarification/Extensions Bootp – RFC 2475: DiffServ Architecture – RFC 2597: DiffServ Assured Forwarding (AF) – RFC 3247: DiffServ Expedited Forwarding (EF) – RFC 3260: New Terminologies and Clarifiations for DiffServ **IPv6** – RFC 2460: IPv6 Specifications – RFC 2461: IPv6 Neighbor Discovery – RFC 2463/4443: ICMPv6 – RFC 2767: Dual Stacks IPv4 & IPv6 – RFC 3315: DHCPv6 (client/relay) – RFC 4291: IPV6 Addressing Architecture
LAYER 2	– IEEE 802.1Q/p: VLANs/Priority – IEEE 802.3ad: Link-Aggregation (LACP)
MANAGEABILITY	– RFC 1157/1902: SNMPv1/v2c – RFC 1213: Management Information Base – RFC 1305: NTPv3 – RFC 5905: NTPv4 – RFC 1591: DNS (client) – RFC 4250/51/52/53/54: SSH (SecureShell) – TACACS/TACACS+
SECURITY	– RFC 2818: HTTP over TLS
OTHERS	– Cloud Native Computing Foundation

Support

- How to reach SnapRoute?
  - http://www.snaproute.com