Red Hat

Redefining carrier grade for service providers

Cloud-native technology is redefining carrier grade with improvements in performance, the ability to scale on demand, and increased agility for new service deployment

The changing nature of telco networks and their impact on carrier grade

Carrier grade is expected to provide a level of availability, or quality of experience, to an end user. Typically, this expectation is an uninterrupted level of service, often with a guarantee or service level agreement (SLA) of 99.999% reliability, as a percentage of availability or uptime. This 99.999% also equates to no more than 5.26 minutes of downtime per year.

To provide a carrier-grade experience, service providers need a robust and resilient network that can deliver the highest possible uptime. This was achieved with the use of purpose-built hardware and physical network elements. Typically, an individual network element type (e.g., router, switch, firewall) would be sourced from a single vendor, tightly integrated with other network elements and fully tested for interoperability prior to deployment. Although cumbersome and complex, these proprietary networks were predictable in terms of what they provided, and their design set a best practice of how a service provider's network should be built and operated. However, this best practice eventually led to the inability to adapt and change to differing requirements and new technologies.

As technology has evolved, and end user services have become more diverse, so have service providers' networks. Networks are no longer purpose-built or standalone. They are modular and cloud technology-based, formed from interoperable building blocks from a variety of vendors. They are typically built using commercial-off-the-shelf (COTS) hardware to realize the most efficient cost and to provide the most flexibility to meet ever-changing demand.

As important as flexibility is, service providers also gain access to a plethora of additional benefits that include increased performance, the ability to scale more efficiently to meet changing demand, and added agility for faster time to market for new services and applications that can take advantage of the network. Service agility is an important benefit, as service providers strive to find different ways they can innovate to differentiate themselves in a highly competitive market. However, they still need to ensure a carrier-grade experience.

To deliver a carrier-grade experience, a service provider's network requires robust capabilities, specifically a level of inherent quality and reliability. Compared to a traditional network where availability was key to the provision of a carrier-grade experience, cloud technology-based networks must also meet other requirements to deliver a carrier-grade experience the cloud-native way.

Cloud-native technology has changed the notion of availability—it is now determined by a combination of different factors

What does it take for a cloud-native service provider network to be carrier grade?

A cloud-native foundation will ensure service providers can deliver the requirements that define a carrier-grade experience beyond availability of the network to include performance and scalability, security, manageability, stability, and sustainability.

Availability

Cloud technology-based network function virtualization infrastructures (NFVI), microservices, and container-based environments rely on high-volume, commodity hardware rather than specialized and purpose-built architectures. For this reason, the concept of service availability is particularly relevant

facebook.com/redhatinc
@RedHat
linkedin.com/company/red-hat

to defining carrier-grade specifications for these environments and is determined by a combination of networking, virtual machine (VM), pod, application availability, fault-detection mechanisms, and accuracy from precise timing mechanisms.

Performance and scalability

Software is key to achieving the necessary levels of performance to meet stringent and changing demand

Similar to traditional purpose-built networks, performance within a cloud technology-based service provider network is somewhat dependent upon the underlying hardware. However, overall performance is mainly dependent on:

- Platform software - should not compromise performance by consuming unnecessary resources as it abstracts the underlying hardware capabilities
- Application software - should be refactored to ensure optimal performance and to take full advantage of cloud-native capabilities

The software environment must be able to support deterministic workloads, offering predictable performance and latency that meets stringent and changing demand. One of the key features of cloud technology-based environments is the ability to dynamically scale inline with demand. The platform must have consistent and efficient scaling capabilities across all aspects of the infrastructure, including compute, storage, and networking.

Security

Security implementation must be an integral part of the overall life cycle

Cloud technology-based networks generally add more complexity, open new attack vectors, and can increase security risks. To control and defend a cloud technology-based system and its active applications, service provider networks need a hardened platform with an integrated, layered security focus. The platform needs to apply continuous checks throughout the entire application life cycle. To ensure security as an integral part of the life cycle, service providers are adopting a development, security, and operations (DevSecOps) approach. This modern methodology prevents security from being isolated to a specific team in the final stages of development. In the collaborative framework of DevSecOps, security is seen as a shared responsibility that is built into initiatives.

Manageability

Life cycle management must ensure the manageability of all software components that may coexist

Manageability covers the key aspects for servicing and maintaining the service provider's network functions and applications. In the cloud domain, this is referred to as life cycle management. One key challenge is the number of different software components that may coexist on the same platform. A suitable life cycle management process or mechanism is responsible for the creation, startup, upgrade, shutdown, and removal of all these software components or applications with minimal service interruption and to avoid the overcomplexity of the environment.

Stability

Cloud-native methodologies promote continuous integration and continuous delivery of software which must not affect stability

Software stability is a key tenet of carrier grade, but could be compromised due to the increased velocity of updates and upgrades that are commonplace with cloud-native environments. Applying the latest software within a service provider's production environment is key to preventing vulnerability exploits and cyberattacks. A cloud-native methodology that encompasses continuous delivery and continuous integration (CI/CD) and that uses platform capabilities, such as canary deployment strategies, will mitigate the stability risk inherent in software updates and upgrades and maintain the carrier-grade experience.

Service providers will use cloud-native technology and new approaches to meet their sustainability goals

Sustainability

Service providers use technology to reduce power consumption and support their own or mandated sustainability goals. With the use of granular hardware control, energy-aware schedulers and autoscalers, and artificial intelligence and machine learning (AI/ML) for smart workload placement, service providers can achieve energy optimization within their network at the node, cluster, system, and domain dimensions. Domain-level optimizations can provide the most significant impact for service providers as they have a relatively high level of control over them, with the radio access network (RAN) accounting for 75% of the total power consumption of their overall network.

Platform-level, cloud-native, carrier-grade capabilities

Operating system

The platform operating system (OS) provides many of the carrier-grade requirements for telecommunications (telco) networks. It has to connect the underlying infrastructure and resources to form an orchestrated environment for service providers to confidently and consistently run their workloads across private and public clouds.

Red Hat® Enterprise Linux® plays a key role in addressing **security**. Its built-in security features and profiles augment the integrated Security-Enhanced Linux (SELinux) that helps service providers define access controls for applications, processes, and files on the system. SELinux provides an additional layer of security and should be enabled continuously. The platform must provide an established and consistent security baseline for service providers to have the confidence to build, run, and scale their workloads within any of the cloud environments they choose to operate. Red Hat Enterprise Linux mitigates risk with integrated and automated controls, ensuring security capabilities are integral throughout the entire life cycle. Automated security configurations maintain consistency at scale and allow service providers to implement and manage security best practices with fewer resources. Red Hat Enterprise Linux centrally manages identities and configures role-based authentication and authorization controls across the entire telco infrastructure.

Red Hat Enterprise Linux's real-time kernel and tuning capability plays a key role in delivering **performance** and **availability** for certain functions. One example is radio access network (RAN) distributed unit (DU) workloads where deterministic precision time protocol (PTP) and process and thread scheduling behavior is important. Specific tuning profiles can be created that are tailored for a particular cloud environment or provider. A node tuning operator is used to achieve low latency **performance**, with unified **manageability** to ensure the greatest flexibility for different application requirements. A tuning profile can be created for the real-time kernel.

Kernel system tuning vastly improves determinism, with thorough system tuning improving consistency of results by approximately 90%. The real-time kernel and system tuning is most appropriate for telco workloads that require stringent high **performance** and low latency for core kernel features, including interrupt handling and process scheduling in the microsecond range. Tuning capabilities give service providers the ability to change attributes of threads (scheduling policy, scheduler priority, and processor affinity) and interrupts (processor affinity).

Regarding **manageability** and **availability**, Red Hat Enterprise Linux uses analytics to proactively identify and fix issues to maximize uptime. Telco cloud deployments are continuously analyzed, with Red Hat Enterprise Linux able to predict risk and recommend actions that help service provider's operational teams prioritize and focus when and where it is needed most.

Red Hat Enterprise Linux CoreOS is a default OS for Red Hat OpenShift®. Red Hat Enterprise Linux CoreOS uses the same quality, **security**, and control capabilities that are used in Red Hat Enterprise Linux but is designed to be immutable and managed in a tighter manner with minimal modification of system settings. Red Hat Enterprise Linux CoreOS also uses a different container engine that has a smaller footprint and reduced attack surface, which facilitates specific compatibility with different Kubernetes versions.

Application platform

A common application platform provides the necessary capabilities to deliver carrier-grade requirements

Many service providers are selecting an enterprise-ready Kubernetes solution with a choice of deployment and consumption options to meet the needs of their business. Red Hat OpenShift gives service providers a unified application platform to enhance business innovation and modernize their applications and infrastructure and to ensure they offer a carrier-grade experience. In conjunction with the OS, the application platform also delivers many carrier-grade requirements.

Red Hat OpenShift allows service providers to build their carrier-grade, cloud-native networks. Red Hat OpenShift is optimized for containers and built on top of Red Hat Enterprise Linux CoreOS, taking advantage of the inherent **security** capabilities within Red Hat Enterprise Linux. As security extends across all aspects of the application life cycle, from development to deployment and operations, service providers need a simplified way to enforce security and compliance with minimal delays or risk to protect their cloud-native workloads. Red Hat OpenShift has out-of-the-box security capabilities that fully support the DevSecOps methodology. Red Hat OpenShift also integrates security into CI/CD pipelines that can detect issues such as vulnerabilities and noncompliant configurations, resulting in higher **availability**. With Red Hat OpenShift's secure-by-default settings and other tools that include the Red Hat Universal Base Image (UBI) and enterprise image registry and scanner, providers eliminate risks and enforce security and governance controls.

Service providers must also ensure they achieve operational consistency and efficiency at scale. A telco cloud that extends to the very edge of the network, and uses a mix of private and public clouds, can create inconsistencies and add complexity. Red Hat OpenShift provides service providers with centralized **manageability** capabilities that include checks for failing components, misconfigurations, policy and compliance, and image scans to streamline their operations. When used with Red Hat Advanced Cluster Management for Kubernetes, Red Hat OpenShift allows service providers to manage their services across clusters through the entire application life cycle with a single deployment view, as well as distributing policies at scale.

Performance is achieved using specific features and capabilities of the application platform to increase throughput, reduce latency, and enhance CPU and storage usage. Some of these features and capabilities include:

- CPU resource guarantees
  - CPU pinning
  - Non-uniform memory access (NUMA) alignment
- Bandwidth guarantees
  - Rate limiting
  - Minimum bandwidth
  - Queuing

- Quality-of-service (QoS)
- Software datapath acceleration
  - Open vSwitch (OVS) with data plane development kit (DPDK)
- Hardware datapath acceleration
  - Smart network interface cards (SmartNICs)
  - Single root input/output virtualization (SR-IOV)
  - Virtual graphics processing units (vGPUs)
  - Field programmable gate arrays (FPGAs)
  - eASICs
  - OVS hardware offload
- Timing guarantees
  - Latency
  - Jitter
  - Precision time protocol (PTP)
  - Synchronous ethernet (SyncE)
- Crypto acceleration
  - Internet protocol security (IPSec)
  - Transport layer security (TLS)
  - Offload capabilities

As an example, Red Hat OpenShift allows service providers to use both SR-IOV and DPDK capabilities. SR-IOV and SmartNICs allow service providers to optimize their platform by allowing multiple functions and applications to run simultaneously. SR-IOV allows the SmartNIC to be shared, as well as gives direct access to the network. DPDK is used to accelerate network traffic.

Cloud-native carrier grade must be by design

Software vendors should adopt cloud-native best practices for application design to ensure a carrier-grade experience

VNFs represented an important step in the journey to greater agility, both for service providers and the wider ecosystem of independent software vendors (ISVs). VNFs made it possible to replace inflexible purpose-built infrastructures, making innovation and the ability to deliver services faster attainable. These early VNF-based networks ran in vertically integrated stacks that simplified relationships with a single vendor. However, similar to proprietary appliances, using a single vendor limited service providers to that vendor’s roadmap and a vertically integrated stack, which over time did not increase agility or lower costs.

Moving beyond simple VNFs to a cloud-native design will achieve a new level of efficiency and agility needed to rapidly deploy innovative, differentiated offers that service providers’ customers demand.

Best practice for cloud-native application design

The most optimal cloud-native application design is with collections of microservices that use container-based technology. Applications need to be built or, often, refactored from their previous incarnations to be considered cloud-native in their design. Several key aspects or attributes of a cloud-native application design approach should be considered to meet the requirements of carrier grade:

- An application built in a cloud-native way will comprise a group of self-contained microservices, each with a single purpose and all of which interact over clearly defined APIs. This ensures they are predictable in terms of **performance**—can be reused, replaced, and upgraded. Using container-based technology makes certain the microservices that constitute an application are packaged with everything required to operate, including libraries and application-specific dependencies, all within a dedicated and isolated space for the utmost **security** capabilities.
- Exposing telemetry data allows service providers to observe and manage the performance and health of the infrastructure and applications. A cloud-native service provider's network needs active observability capabilities rather than passive monitoring tools so that data sources are able to contribute to additional analytical processes that can represent the overall state of the system. A cloud-native network also facilitates a separate entity within the system for collecting and correlating information and logging data, which increases the **manageability** of the system overall.
- An application must have full life cycle management, including the ability to be instantiated, configured, scaled, upgraded, stopped, and deleted. The application must respond to platform events, in terms of initialization, to accept services and resource release prior to shutdown, along with the ability for continuous and in-service upgrades where multiple application versions are run in parallel accepting differing traffic amounts to prove **performance** and stability before going fully into production.
- The cloud's inherent constraints and reliability characteristics are fully adhered to by a cloud-native design, whereby failure of an individual component or microservice will not cause the overall **availability** of the system to fail. Cloud-native applications should fail fast, being readily replaced with minimal loss of **performance** and **availability**. A key requirement for this capability is that the microservices cannot contain state data that would be lost upon failure. Persistent volumes should be used to store files or block level data, with an external low latency, key-value store used to save user session information. Maintaining critical functionality in the application platform can be achieved using replica pods that ensure a specific number is running in the system at any given time to guarantee **availability**.
- A cloud-native approach allows for **scalability**. When an application is under heavy demand, this approach creates new instances and subsequently removes them once the demand has been served.
- The internal architecture of an application should adhere to modern cloud-native design principles. Applying those principles to a service provider framework mandates the existence of an ingress tier (e.g., load balancer). This is realized by a set of microservices that provide service discovery and exposure for the service provider CNFs that are highly distributed and scaled by nature. State handling is also a critical design aspect of CNFs due to the hybrid nature of service provider CNFs

processing stateful components and modernized stateless microservices, which necessitates an application tier for processing stateless microservices and a data store tier for storage and to ensure latency and **performance**.

Validating and certifying partner cloud-native applications

A rigorous and comprehensive vendor application validation and certification process will give service providers the confidence their chosen applications can be fully integrated into their application platform

Even though cloud-native application design principles are well documented, their increase in popularity has proliferated the unregulated and simple migration of existing applications. In some cases, this approach has resulted in unacceptable levels of **performance, security,** and interoperability with the platforms in which they were deployed and between different vendor's applications. Red Hat strives to ensure CNF certification is rigorous, comprehensive, and dependable for service providers and ISVs.

Red Hat gives service providers the confidence their chosen CNF partner vendor applications can be fully integrated into their Red Hat OpenShift application platform.

Table 1: Partner options for certifying CNF solutions

	Vendor validated	Certified CNF for Red Hat OpenShift
Support	• Continuously tested • Collaborative support	• Continuously tested • Collaborative support
Platform integration	• Runs on Red Hat OpenShift	• Engineered with Red Hat • Integrated in Red Hat OpenShift
CNF verification	• Validated by vendor	• Verified by Red Hat
Base image	• Red Hat OpenShift compatible base image	• Built on Red Hat Enterprise Linux or Universal Base Image (UBI)

Vendor validated

This capability offers interoperability verification of a CNF with Red Hat OpenShift by the application vendor, in collaboration with Red Hat. The vendor commits to offer commercial support for the CNF on Red Hat OpenShift, backed by a collaborative support framework with Red Hat. The partner implements and operates a CI/CD environment and tests its CNF with Red Hat OpenShift, as changes in either product are developed and released.

Certified CNF for Red Hat OpenShift

This certification sets the highest standard for critical network functions on Red Hat OpenShift and extends the collaboration between Red Hat and its partners to apply best practices specific to telco deployments. The partner performs and passes a set of CNF-specific tests in a CI/CD environment as changes in either product are developed and released. This delivers deeper integration with Red Hat OpenShift and is engineered with Red Hat, taking advantage of Red Hat Enterprise Linux as a trusted container foundation that includes the operator framework to provide ongoing interoperability, **security**, maintenance, and life cycle alignment, as well as automated management of the CNF life cycle.

Red Hat partners with industry-leading hardware, software, and cloud vendors to offer more choice, innovation, and stability. The company fosters a large certified partner ecosystem and is certified for use with all major cloud providers. Service providers can build, deploy, manage, and secure applications on Red Hat OpenShift with certified Kubernetes operators and Helm charts, and extend Red Hat OpenShift with their preferred tool sets from both Red Hat and a growing list of certified operators from hundreds of partners.

Conclusion

A cloud-native service provider's network needs to possess a level of robustness and reliability to deliver the availability, performance and scalability, security, manageability, stability, and sustainability requirements that define the term carrier grade. These requirements are necessary to meet stringent service and user experience expectations, relying upon the OS, security, performance, and operational capabilities at the platform level.

Red Hat OpenShift gives services providers an open cloud-native application platform that can support any workload on any footprint at any location. Red Hat OpenShift's rich set of capabilities coupled with a verified and certified ecosystem of network functions and applications will ensure service providers have a reliable cloud-native foundation to deliver a carrier-grade experience.

Learn more

Red Hat OpenShift Container Platform

Red Hat Enterprise Linux

Red Hat Advanced Cluster Management for Kubernetes

Red Hat's ecosystem catalog

Red Hat Universal Base Image (UBI)

About Red Hat

Red Hat helps customers standardize across environments, develop cloud-native applications, and integrate, automate, secure, and manage complex environments with award-winning support, training, and consulting services.

facebook.com/redhatinc
@RedHat
linkedin.com/company/red-hat

North America
1 888 REDHAT1
www.redhat.com

Europe, Middle East, and Africa
00800 7334 2835
europe@redhat.com

Asia Pacific
+65 6490 4200
apac@redhat.com

Latin America
+54 11 4329 7300
info-latam@redhat.com