Implementing Service Portability at the Edge with Open Source

Sponsored by: WIND

WIND™

WHEN IT MATTERS,

IT RUNS ON WIND RIVER.

For nearly 40 years, Wind River software has enabled digital transformation in telecommunications and other critical infrastructure sectors –and we're just getting started.

Find out how our open solutions will help you deploy faster, lower operating cost, and deliver the performance, security, and reliability your customers expect.

FIND OUT NOW

Building Cloud-Native Services: Not Exactly Like Cloud Apps

Everyone's familiar with cloud native apps, but cloud native services are newer and feature unique content, although common best practices apply to building both.

By Derek Handova

Cloud-native methodologies have been discussed extensively when it comes to building cloud-native applications. In contrast, cloud native services have been written about much less frequently, which is why this article is necessary to clarify what they are, how they differ from cloud-native apps, and why they're important for edge architecture. It comes down to what is on the other end of cloud-native technology. If it's the user, such as in the chat use case, then it's an application. But if it's another program, like when that chat app interacts with an online database such as Amazon's DynamoDB or Azure SQL to store messages and chat logs, then it's a cloud native service.

"If the user of the software is a consumer—a human being—it is an application," said Neeraj Murarka, CTO and co-founder of Bluzelle, provider of a decentralized database protocol. "If the user is another piece of software interacting with the software on the cloud, the latter is a service. In this context, an application is likely a primary user of the service."

In any event, there are common best practices for both use cases. But what's most important is how a cloud-native mindset contributes to developing services in an edge-centric architecture. That's because the definition of cloud-native services continues to evolve, which means there must be continuing evolution of service integration and delivery.

"If we build a cloud-native application from scratch, we need to customize and integrate it correctly into our IT infrastructure," said Andrei Lipnitski, ICT department manager atScienceSoft, provider of IT consulting services and custom software development. "When building cloud-native services, the best practice is to use continuous integration and continuous delivery (CI/CD) as a building and test environment."

Differences in Building Cloud-native Apps, Services

Both cloud-native applications and cloud-native services should be built for portability. However, when building a cloud-native application, overall user experience (UX) is a key focus. With cloud-native services the key focus is making sure the service can serve an array of apps.

"Not only do you have to ensure functionality of each component and interaction via APIs but also that the end product provides a great UX," said Miha Kralj, managing director of cloud strategy at Accenture. "When creating a cloud-native service, ensure that the functionality of your service can serve a variety of applications, and that the right APIs are exposed, at the right level, to give maximum capability, performance, resilience, and scale."

While cloud-native applications have to be completely portable without any dependencies on cloud-provider-specific APIs, cloud-native services can be dependent on the cloud platform such as an AWS Lambda function, according to Eran Bibi, head of DevOps at Aqua Security, platform provider for securing cloud native applications.

"While the app it serves may be portable, the function will need to be reconfigured to run elsewhere," said Bibi. "While a single service may not be limiting, using many will create dependencies that limit application portability."

Common Best Practices for Cloud Native Apps, Services

With both cloud-native applications and services, developers can enjoy the freedom to craft microservices loosely coupled to their chosen language—but there are limits.

"It is advisable to create guidelines for how to build containers to achieve reusability of components," said Manuel Nedbal, founder and CTO at ShieldX Networks, a cloud security provider. "Start with a clear understanding of what constitutes a microservice and which functional blocks each encompass. Without those practices, it is hard to assemble individual microservices into an application and foster component reuse."

Other common best practices for building cloud native applications and services include:

- Architect for failure: Assume cloud infrastructure and services will not always be available, according to Nitzan Miron, VP of application security services product management at Barracuda Networks. "Follow the vendor's recommendations, deploy resources to multiple regions, and fail gracefully when resources become unavailable," he says.
- Dynamically allocate all resources: According to Lee Atchison, senior director of cloud architecture at New Relic. "Build your application to allocate resources on the fly when required. This way, you create applications that do not become underpowered and underperforming in high load environments," he says. "Rather, they scale better under peak load yet save cost of unnecessary idle resources during low load times."

Cloud-Native Mindset for Edge Computing Portability

Building for edge-computing portability requires a different mindset. Edge-centric architecture needs to account for the compute, data collection, and communication aspects and must be able to allocate more intelligence to distributed nodes and edge devices.

"This is where cloud native mindset combined with the maturity and availability of integrated services that can handle ingest, collection, processing, storage, and analytics in different cloud platforms helps in architecting effective solutions," said Abay Radhakrishnan, CTO and architect at Sungard Availability Services.

A cloud-native mindset is key to leveraging compute at the edge. Cloud-native applications are small and stateless, which increases application mobility and scalability, whereas a service will span the data center, edge cloud, far edge, and device at the edge, according to cloud experts.

"All these components are brought together to deliver the service," said Gareth Noyes, Wind River chief strategy officer. "The advantage of having common technology for IoT and [operational technology] is that you can take cloud native applications and deploy at the cloud, fog, and edge levels."

The rapid rise of containers in telecommunications. But what about VMs?

By Charlie Ashton

Despite the rapid evolution in virtualization over the past couple of years, service providers have concluded that even more drastic advances are needed. In order to meet their long-term goals of delivering compelling end-user experiences from highly-efficient operations that are based on modern technologies pioneered by the webscale giants, service providers are turning their focus toward "cloudification" as the next strategic initiative beyond virtualization.

Since the introduction of Network Functions Virtualization (NFV) in 2012, we've seen physical network functions (PNFs) being replaced one by one with virtual network functions (VNFs) via virtual machines (VMs.) In contrast cloudification is implemented through the deployment of "cloud-native" services, Cloud-native services are developed from the ground up for deployment in cloud or edge data centers. Designed for simplicity, flexibility and rapid deployment, cloud-native services are typically implemented as microservices that run within Linux containers orchestrated by Kubernetes, rather than as complex VNFs that run within VMs.

A cloud-native strategy allows service providers to accelerate both the development and deployment of new services, while the ability to rapidly scale up or scale down those services allows for resource utilization to be optimized in real-time, in response to traffic spikes and one-time events.

Cloud-native services are also key to "Zero-Touch Automation" (ZTA), which is the term used to describe the automation of IT and data center infrastructure. In the telecom industry, ZTA extends this concept of automation beyond the initial installation phase to cover the entire lifecycle of network operations including planning, delivering, onboarding, monitoring, updating and ultimately, decommissioning of services. ZTA will move telecom networks from today's automatic functions to fully-autonomous operations that bring

significant top-line revenue improvements as well as sustainable reductions in operational costs.

As a leading provider of software infrastructure platforms for telecom networks, Wind River has been collaborating closely with service providers as they refine their strategies for cloud-native deployments. It's clear that while service providers continue to launch new network services based on VNFs running in VMs, they are also actively starting to explore the deployment of additional cloud-native services that are based on containers.

While many applications will be implemented as containers in line with this cloud-native approach, others will remain as VMs, so it's important that the underlying virtualization infrastructure treats both as first-class citizens, co-existing in the telecom cloud.

Wind River Titanium Cloud, a secure software infrastructure platform, supports containers in addition to VMs. For maximum flexibility, containers can run either within VMs for maximum tenant isolation and security, or on "bare metal" for minimum footprint. This bare metal container support is currently characterized as a "tech preview" function as we partner with key customers to confirm exactly what level of functionality will be required for their near-term use cases. Future functionality will be upstreamed to the StarlingX edge compute infrastructure project hosted by the OpenStack Foundation.

While Titanium Cloud addresses the needs of containers in cloud data centers and edge infrastructure, service providers see additional business advantages in deploying container-based applications in network appliances as well as specific network subsystems such as the Radio Access Network (RAN). Wind River addresses these use cases through containerized solutions based on Wind River Linux.

The recent introduction of Wind River Linux LTS 18 provides applications, tools, documentation and other resources for embedded system developers looking at leveraging or deploying systems using a cloud-native model. Wind River Linux now includes pre-integrated components from the Cloud Native Computing Foundation (CNCF) configured to deliver a fully-functional solution for embedded systems such as edge appliances. By leveraging this pre-integrated container platform, developers are able to focus their time on creating the applications that represent their true differentiation, enabling them to accelerate their time-to-market for value-added solutions.

At Wind River, we're delighted to be contributing to the realization of the cloud-native concept within telecom infrastructure. If you'd like to know more about Titanium Cloud or Wind River Linux, please browse the information available online or contact Wind River to arrange a face-to-face discussion.

Sponsored by: WIND

Why Open Source Is Important for Service Portability

Service portability in edge computing depends on top open source platforms, but you have to know how they work together—and how they don't.

By Derek Handova

Service portability is a major concern with new edge-centric architectures. Operators need the flexibility to apply cloud-native principles to shift services between a device, local cluster, and the cloud as they scale services at the edge.

Open source is often looked at as a key component of service portability, but is it essential in edge-computing use cases? If so, why is it considered essential? Moreover, with open-source projects continuously proliferating, operators need to know how these projects can and can't work together.

"The recent announcement of the LF Edge community by the Linux Foundation not only speaks to importance of open-source contributions to edge computing but also reinforces how very fragmented our edge landscape remains," said Matt Caulfield, formerly principal engineer of edge computing at Cisco and now CEO of Oort, which helps enterprise adoption of edge computing.

While edge-computing use cases have yet to converge on a standard set of open-source platforms, some experts believe containers and a Kubernetes-like platform are the future of edge-centric environments and service portability. Open-source platforms that tightly integrate together and could provide orchestration across container and VM technologies, such as Open Network Automation Platform (ONAP), will help operators realize service portability.

"Projects such as Kubernetes and OpenStack with StarlingX can leverage other open-source

projects through abstraction layers," said Gareth Noyes, chief strategy officer at Wind River. "For instance, Ceph or other open-source storage backends can be leveraged by Kubernetes through storage drivers and

with OpenStack through backend storage drivers. Edge deployments could benefit from tighter integration, more sharing of resources, and less duplication of functionality to make more efficient use of resources at the edge."

Open Standards are the Building Blocks for Service Portability

Open standards-based technologies like LXD, a Linux container management system kernel-based virtual machine (KVM), and Open Source NFV Management and Orchestration (OSM) form the basic building blocks that enable service portability in edge-centric environments.

"Edge computing remains an area with a high degree of innovation that open source is uniquely positioned to address," said Stephan Fabel, director of product for Canonical, the company created alongside the open source platform Ubuntu. "The building blocks for open-standard implementations across any industry are open source, which is visible in standards put forth in working groups within ETSI and other bodies."

While open source enables innovation in edge computing, security concerns remain. For operators to feel confident that open-source platforms can safely support service portability at the edge, open-source platforms must adhere to security protocols and regulatory requirements. Therefore, open source must be able to be audited to ensure it can provide an acceptable service level and is not vulnerable to attack, according to other experts.

"As more and more enterprises want to benefit from edge-centric computing, they will also need to go through the process of ensuring the platform is safe and suitable for their needs," said Neeraj Murarka, CTO and co-founder at Bluzelle, a decentralized NoSQL database supporting JavaScript. "Open source gives these companies the transparency they need to perform this analysis to whatever level they require."

How Open Source Projects Do and Don't Work Together

When it comes to open-source projects working together for service portability, platforms like StarlingX, ONAP, and OpenStack combine multiple components such as Ceph and Kubernetes into a full platform for the distributed edge.

"The key benefit is the integrated approach to manage complex aspects like orchestration, configuration management, storage, and networking at the edge," said Greg Luck, CTO at Hazelcast, an operational data management company. "While integration, auto-provision, and orchestration are offered by many open-source platforms, common identity and access management and policy management varies across the platforms limiting service portability."

Open-source orchestration projects like ONAP promote a lightweight, developer-centric orchestration and marketplace model to facilitate interoperability. Projects like StarlingX, enable services for a distributed edge cloud. StarlingX builds on existing services in the open-source ecosystem by taking components of projects like Ceph, OpenStack, and Kubernetes and complementing them with new services such as configuration and fault management, according to Shamik Mishra, assistant vice president of technology at Aricent, a design and engineering company.

"Scale and complexity is a key problem that needs to be solved, and NFV/SDN and telco-centric orchestration systems are not exactly friendly to edge application developers used to three-click-run cloud-computing," Mishra said. "These developers don't understand the complexity of a carrier network, so some of the open-source orchestration projects like ONAP may have to think of a lightweight developer-centric orchestration and marketplace platform."

As with all projects, it's important to understand what you are trying to accomplish. Then select the tools that best align with your desired outcomes.

"Spanning workload orchestration, distribution, storage, and high availability at the edge, each of the above is purpose-built to help organizations execute cloud-native and edge-centric strategies at scale," said Jeff Burk, senior VP of engineering at Collibra, a data governance platform.