International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 5 Issue 5, July-August 2021 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Deploying and Managing Containers on RHEL with Podman and Buildah

Michael Ondaatje, Yann Martel

Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada

ABSTRACT

As containerization becomes the cornerstone of modern application deployment, Red Hat Enterprise Linux (RHEL) offers robust, secure, and flexible tools to manage container lifecycles without the need for a traditional daemon. This article delves into the powerful capabilities of Podman and Buildah, two open-source tools that redefine container management on RHEL by enabling daemonless, rootless container operations with enhanced security and compatibility. It explores the architectural differences from Docker, practical deployment workflows, and best practices for building, running, and maintaining containers in enterprise environments. Through detailed insights and real-world examples, the article demonstrates how organizations can leverage Podman and Buildah to streamline container workflows, improve security posture, and achieve seamless integration with existing RHEL infrastructure—empowering teams to build scalable, portable, and compliant containerized applications with confidence.

How to cite this paper: Michael Ondaatje | Yann Martel "Deploying and Managing Containers on RHEL with Podman and Buildah" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-5 | Issue-5, August 2021, pp.2522-2529, URL: www.ijtsrd.com/papers/ijtsrd45001.pdf

IJTSRD45001

1. INTRODUCTION

Containerization has revolutionized modern application development and deployment by enabling lightweight, portable, and consistent runtime environments. Containers encapsulate applications and their dependencies, ensuring that software runs reliably across diverse computing environments—from developer laptops to on-premises servers and cloud platforms. This paradigm shift has accelerated innovation cycles, improved resource utilization, and simplified application scaling.

At the heart of many enterprise container strategies is Red Hat Enterprise Linux (RHEL), a trusted, secure, and high-performance operating system designed to support critical workloads. RHEL's stability, extensive ecosystem, and compliance with enterprise standards make it an ideal platform for deploying containerized applications. To harness the full potential of containerization on RHEL, administrators and developers require tools that seamlessly integrate with the OS while enhancing security and operational flexibility.

Podman and Buildah have emerged as pivotal open-source tools within the RHEL ecosystem that address these needs. Unlike traditional container runtimes that rely on a central daemon, Podman operates in a daemonless, rootless mode, significantly improving security by reducing attack surfaces and granting users more granular control. Buildah complements this by providing efficient, scriptable image-building capabilities that integrate natively with RHEL's architecture.

This article aims to provide a comprehensive guide for deploying and managing containers using Podman and Buildah on RHEL. Readers will gain insights into the fundamental concepts, practical workflows, and best practices for leveraging these tools to build, run, and maintain containerized applications in enterprise environments. Whether you are an IT administrator, developer, or DevOps professional, this guide will equip you with the knowledge needed to streamline container management on RHEL and enhance your organization's cloud-native journey.

2. Understanding Containerization and Its Benefits

Containerization represents a fundamental shift in how applications are developed, packaged, and deployed, differing significantly from traditional virtualization methods. Unlike virtual machines (VMs), which emulate entire operating systems including the kernel, containers share the host OS kernel while isolating application processes. This architectural difference makes containers inherently more lightweight and efficient.

Containers vs. Traditional Virtual Machines

Virtual machines require dedicated guest operating systems, consuming substantial system resources such as CPU, memory, and storage. Each VM is a full OS instance, leading to slower boot times and larger resource footprints. Containers, by contrast, encapsulate only the application and its dependencies within a lightweight, isolated user space on the host OS. This approach enables faster startup times, smaller image sizes, and greater density of workloads on the same hardware.

Advantages of Containers: Lightweight, Portable, Fast Startup

Containers excel in portability—allowing developers to build an application once and run it consistently anywhere, whether on a developer's machine, on-premises servers, or cloud infrastructure. Their lightweight nature enables rapid scaling and efficient utilization of compute resources. Containers start almost instantaneously, which is critical for dynamic, microservices-based architectures where services may need to spin up or down based on demand.

Why Enterprises Prefer Containers on RHEL

Red Hat Enterprise Linux provides a secure, stable, and certified platform for container workloads, aligning with enterprise requirements such as compliance, lifecycle support, and integration with existing IT infrastructure. RHEL's container-focused enhancements, including SELinux policies, container-aware kernel modules, and seamless integration with Red Hat OpenShift, make it a preferred environment for running containerized applications at scale.

Overview of Container Runtimes and Tools Ecosystem

The container ecosystem includes various runtimes and tools designed to build, run, and manage containers. Docker popularized container usage but relies on a central daemon, which introduces certain security and management complexities. In contrast, tools like Podman and Buildah—both natively supported on RHEL—offer daemonless, rootless operations that improve security and flexibility. Complemented by orchestration platforms like Kubernetes and OpenShift, these runtimes form a comprehensive ecosystem enabling efficient container lifecycle management.

This section sets the foundation for understanding why containerization, especially on a robust platform like RHEL, is transforming enterprise IT operations and application development workflows.

3. Introducing Podman and Buildah

As container technology matures, new tools are emerging to address challenges related to security, efficiency, and flexibility in container management. Among these, **Podman** and **Buildah** have gained significant traction—especially within Red Hat Enterprise Linux (RHEL) environments—as robust, open-source alternatives to traditional container tools.

What is Podman?

Podman is a daemonless container engine designed to manage and run containers and pods with a focus on security and simplicity. Unlike Docker, which relies on a central daemon process running with root privileges, Podman operates without a daemon, launching containers as child processes of the command line interface. This architecture eliminates the single point of failure inherent in daemon-based models and allows users to manage containers securely without requiring elevated privileges.

Podman supports many Docker CLI commands and container image formats, enabling an easy transition for users familiar with Docker. However, its ability to run containers rootless enhances security by limiting the potential attack surface and reducing the risk of privilege escalation.

What is Buildah?

Buildah is a specialized tool focused on building Open Container Initiative (OCI)-compliant container images. It allows developers and system administrators to create container images from scratch or using existing images, with fine-grained control over the build process. Buildah supports scriptable image creation workflows that integrate smoothly with automated build pipelines, offering a lightweight and flexible alternative to Docker's image build functionality.

With Buildah, users can manipulate image layers, configure metadata, and optimize builds without running a container engine daemon, making it ideal for secure and efficient container image generation.

Benefits of Daemonless Container Management with Podman

By eliminating the need for a persistent daemon, Podman reduces complexity and enhances system security. Daemonless operation means containers run

under the user's own process tree, simplifying troubleshooting and resource management. This model also enables easier integration with existing system management tools and adheres better to Unix philosophy by avoiding long-running background processes when not needed.

Security Enhancements: Rootless Containers and SELinux Integration

Podman and Buildah bring significant security improvements to container management. Rootless container execution allows users to run containers without requiring root privileges, significantly mitigating risks associated with container escapes or privilege escalation attacks. Furthermore, Podman leverages SELinux (Security-Enhanced Linux) policies native to RHEL to enforce mandatory access controls on containers, isolating them from the host system and other containers more effectively.

This combination of daemonless operation, rootless execution, and SELinux integration makes Podman and Buildah particularly well-suited for enterprise environments where security, compliance, and operational control are paramount.

4. Installing Podman and Buildah on RHEL

Setting up Podman and Buildah on Red Hat Enterprise Linux (RHEL) is a straightforward process designed to quickly enable container management capabilities in your environment. Both tools are officially supported by Red Hat and are included in the standard RHEL repositories, which ensures stability, compatibility, and security.

To begin installation, ensure your system meets the necessary prerequisites: a supported version of RHEL (typically RHEL 7.6 and above), a modern Linux kernel version to support container features, and sufficient disk space to store container images and related data. Network connectivity is also essential to download container images from external registries.

Installation is performed using RHEL's native package management system, which automatically handles dependencies and configurations required for Podman and Buildah. Once installed, you should verify the tools' availability and functionality by checking their versions and conducting basic operational tests. This confirmation step ensures the environment is correctly prepared for container deployment and management.

Configuration considerations include setting up proper user permissions, especially for rootless container management, which allows running containers without elevated privileges, enhancing security. Additionally, enabling and properly configuring SELinux on RHEL is crucial, as Podman and Buildah leverage SELinux policies to isolate container processes and enforce security boundaries effectively.

By following these guidelines, system administrators and developers can rapidly deploy Podman and Buildah on RHEL, laying the foundation for efficient, secure, and scalable containerized application workflows.

5. Building Container Images with Buildah

Buildah serves as a powerful tool for creating container images that comply with the Open Container Initiative (OCI) standards, providing flexibility and control beyond traditional Docker-based workflows.

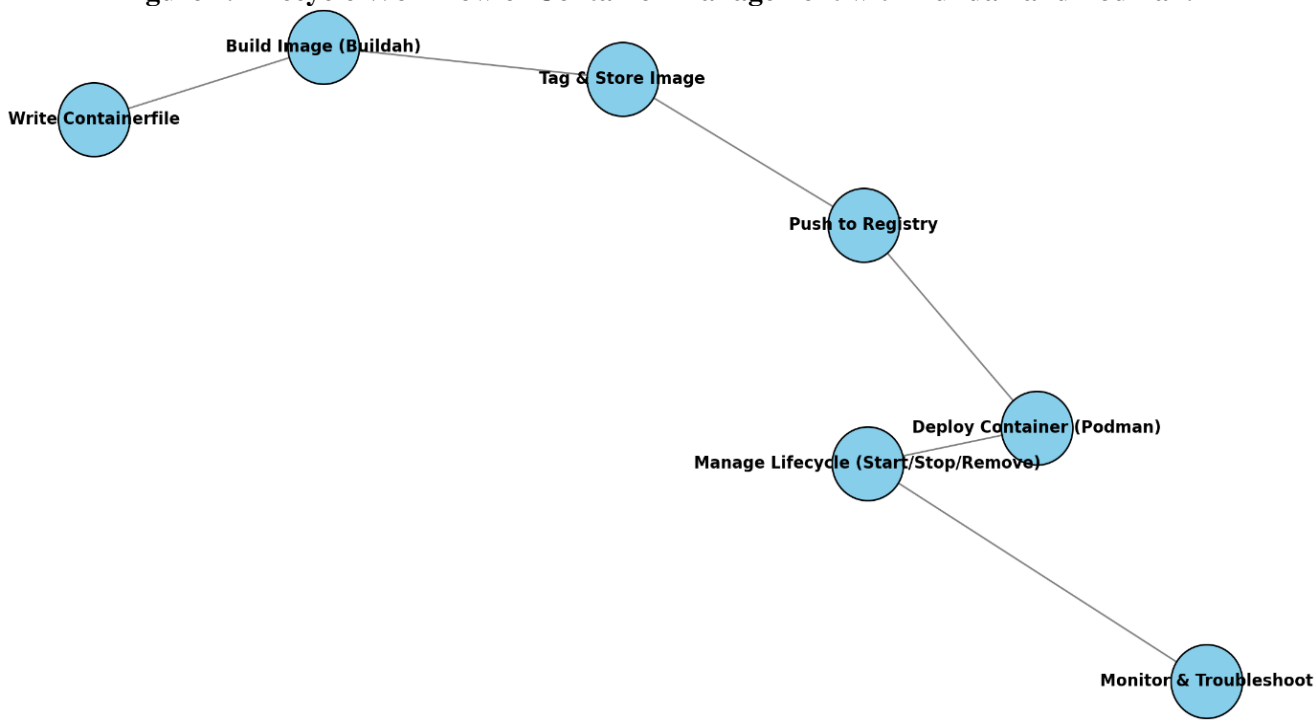
One of the foundational tasks in container image creation involves writing containerfiles—text files that define the instructions for assembling an image. These containerfiles are compatible with Dockerfile syntax, allowing users to specify base images, install packages, configure environment variables, and define startup commands. Buildah interprets these instructions to construct images tailored to application requirements.

Using Buildah's command-line interface, developers and system administrators can build images efficiently. The tool supports both scripted workflows and interactive sessions, enabling granular control over each step of the build process. This flexibility is particularly valuable when optimizing images for size, security, or performance.

Once images are built, they can be tagged with meaningful identifiers to manage different versions and facilitate tracking. Buildah supports storing images locally on the host system, as well as pushing them to remote container registries for distribution and deployment. This ensures seamless integration with container orchestration platforms and continuous integration pipelines.

To maximize efficiency and security during image creation, best practices should be followed. These include minimizing the number of image layers, using trusted base images, regularly scanning images for vulnerabilities, and avoiding embedding sensitive data within the images. Adhering to these principles helps maintain lean, secure images that align with enterprise compliance standards.

Figure 2. Lifecycle Workflow of Container Management with Buildah and Podman.



In summary, Buildah empowers users to craft highly customizable container images with precision and security, making it an essential component of modern container workflows on RHEL and beyond.

6. Deploying and Managing Containers with Podman

Podman provides a robust and flexible platform for running and managing containers on Red Hat Enterprise Linux (RHEL), designed to offer a seamless experience without the need for a centralized daemon. Its compatibility with Docker commands and additional security features make it an ideal choice for container deployment.

At its core, Podman allows users to run containers using straightforward commands and various options tailored to application needs. This includes specifying resource limits, environment variables, and runtime configurations that ensure containers operate efficiently and reliably.

Managing the container lifecycle is fundamental to containerized environments. Podman simplifies this by providing intuitive controls to start, stop, restart, and remove containers as needed. This granular management enables administrators to maintain optimal system performance and ensure that containerized applications are always responsive.

Pods are a distinctive feature in Podman's architecture, allowing multiple containers to be grouped together sharing networking namespaces and storage volumes. This grouping facilitates tighter integration of related containers, simplifying communication between them and improving resource sharing, which is especially useful for microservices architectures.

Networking and volume management are critical aspects of container operations. Podman supports flexible network configurations, including bridge, host, and macvlan modes, enabling containers to communicate internally or externally as required. Volume management ensures persistent storage by mapping host directories or dedicated volumes into containers, preserving data beyond container lifecycles.

One of Podman's most significant advantages is its support for rootless containers, which run without requiring administrative privileges. This enhances security by reducing the attack surface and mitigating risks associated with running containers as root. Rootless mode, combined with SELinux integration, ensures that containerized applications operate with strong isolation and containment.

In summary, Podman equips users with comprehensive tools to deploy, manage, and secure containers effectively, empowering enterprises to adopt containerization with confidence on RHEL environments.

7. Integrating Podman and Buildah in CI/CD Pipelines

In modern software development, continuous integration and continuous delivery (CI/CD) pipelines are essential for accelerating release cycles while maintaining quality and consistency. Integrating container management tools like Podman and Buildah into these pipelines empowers teams to automate the build, test, and deployment processes of containerized applications with greater efficiency and control.

Automating image builds is a critical step in CI/CD workflows. Buildah's lightweight and daemonless architecture allows seamless integration into pipeline stages where container images are created based on application code changes. This automation ensures that new image versions are built consistently, incorporating the latest updates and configurations without manual intervention.

Podman complements this by facilitating container deployments and runtime management within pipeline steps. Automated testing, container startup, and environment validation can be orchestrated using Podman commands, enabling rapid feedback and reducing errors before promoting builds to production.

Popular CI/CD platforms such as Jenkins, GitLab CI/CD, and others readily support the incorporation of Podman and Buildah through scripted steps or plugin integrations. These tools can invoke container build and deployment commands as part of predefined jobs triggered by code commits or merge requests, providing end-to-end automation from source code to running containers.

By scripting container workflows, development teams can define repeatable processes for building, tagging, pushing images to registries, deploying containers to test environments, and rolling out updates to production. This level of automation enhances agility, reduces human error, and enforces consistency across diverse environments.

Moreover, integrating Podman and Buildah within CI/CD pipelines aligns well with modern DevOps practices and Infrastructure as Code (IaC) philosophies, enabling teams to maintain declarative, version-controlled infrastructure and application states. It also improves traceability and auditability, critical for compliance in enterprise settings.

In essence, leveraging Podman and Buildah within CI/CD pipelines empowers organizations to achieve faster, more reliable container delivery, supporting continuous innovation and robust application lifecycle management on RHEL and other platforms.

8. Security Considerations and Best Practices

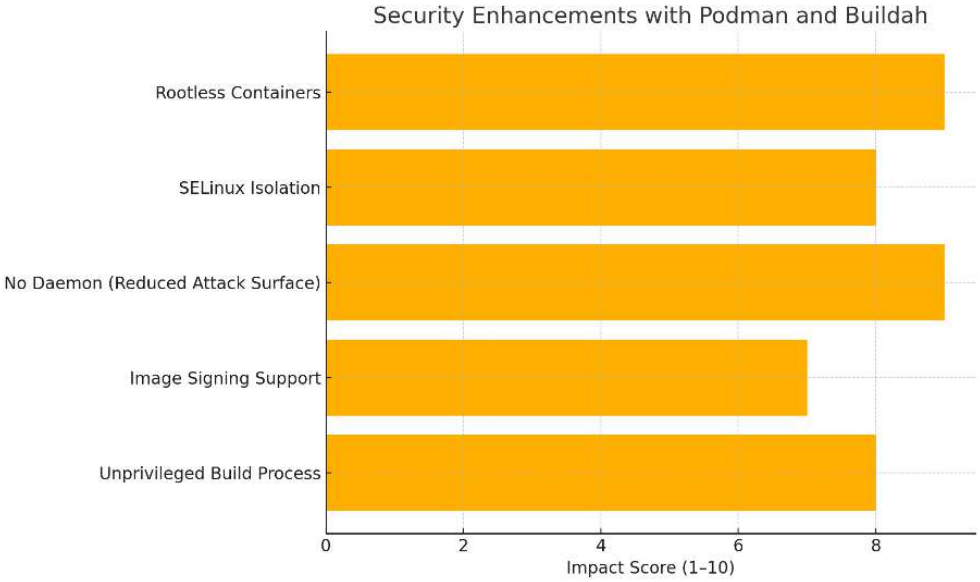
Security remains a paramount concern in containerized environments, especially when deploying and managing containers at scale. Leveraging Red Hat Enterprise Linux's robust security framework alongside Podman and Buildah provides a strong foundation for protecting container workloads.

A key aspect of container security on RHEL is the integration with SELinux (Security-Enhanced Linux). SELinux enforces mandatory access control policies that isolate containers from the host system and each other, limiting the potential impact of a compromised container. Properly configuring SELinux and related system policies ensures that containers operate within tightly defined security boundaries, reducing risks of privilege escalation or unauthorized access.

Managing image vulnerabilities is another critical security practice. Container images should be regularly scanned for known vulnerabilities using security tools integrated into the development lifecycle. Timely updates and patching of base images and application dependencies mitigate exposure to exploits. Automated pipelines that incorporate vulnerability scanning help maintain a secure image supply chain.

Rootless containers, a standout feature of Podman, further enhance security by enabling containers to run without root privileges. This minimizes the attack surface by preventing containers from having unnecessary access to sensitive system resources, thereby reducing the risk of host system compromise in the event of container breaches.

Figure 1. Security Enhancements Achieved by Using Podman and Buildah on RHEL.



To ensure trustworthiness and integrity in container deployments, image signing and verification mechanisms should be employed. Signing images cryptographically confirms their authenticity and origin, while verification before deployment prevents unauthorized or tampered images from entering production environments. This practice is essential for compliance and governance in enterprise settings.

Incorporating these security considerations and best practices into container workflows not only safeguards applications and data but also fosters confidence in container adoption. It enables organizations to maintain resilient, compliant, and secure operations while leveraging the agility and scalability of container technologies on RHEL.

9. Troubleshooting and Monitoring Containers on RHEL

Effective troubleshooting and monitoring are critical to maintaining the health, performance, and reliability of containerized applications running on Red Hat Enterprise Linux (RHEL). Understanding common issues and employing the right tools ensures rapid diagnosis and resolution, minimizing downtime and operational disruptions.

Containers may encounter a range of issues—from startup failures and networking conflicts to resource exhaustion and unexpected crashes. Identifying these problems begins with diagnostic commands provided by Podman, which offer detailed insights into container status, configuration, and error states. These commands enable administrators to quickly inspect container metadata, running processes, and resource allocation.

Podman's event and logging capabilities are invaluable for in-depth troubleshooting. The event stream captures container lifecycle activities, such as creation, start, stop, and errors, providing a chronological view of container behavior. Meanwhile, access to container logs helps surface application-level errors or warnings, essential for pinpointing root causes and understanding runtime anomalies.

Monitoring resource usage is fundamental for performance tuning and capacity planning. Tracking CPU, memory, disk I/O, and network utilization of containers ensures workloads are operating within expected parameters and helps identify bottlenecks or resource contention. Proactive monitoring enables teams to anticipate scaling needs and optimize infrastructure allocation.

To further enhance container observability, a variety of tools and plugins can be integrated with Podman on RHEL. These include metrics exporters, visualization dashboards, and alerting systems that provide real-time visibility into container health and cluster-wide status. Such tools facilitate continuous monitoring and automated responses to emerging issues, aligning with modern DevOps practices.

In summary, combining Podman's built-in diagnostic features with comprehensive monitoring strategies equips administrators with the capabilities needed to maintain robust container environments, ensuring high availability and efficient operations across enterprise workloads.

10. Case Studies and Real-World Usage

The adoption of Podman and Buildah on Red Hat Enterprise Linux (RHEL) has proven transformative for organizations seeking to modernize their application deployment processes, improve security

posture, and accelerate development cycles. Real-world implementations highlight the tangible benefits and lessons learned across diverse industries.

One notable example is a leading financial services company that transitioned from traditional container platforms to Podman on RHEL. Faced with stringent security and compliance requirements, the organization leveraged Podman's rootless container capabilities and SELinux integration to significantly reduce their attack surface. This migration not only enhanced their security framework but also improved agility, allowing development teams to deploy and update applications faster without compromising governance. The shift resulted in streamlined workflows, reduced operational overhead, and improved auditability, which were critical for regulatory adherence in the financial sector.

In the healthcare domain, a major provider utilized Buildah and Podman to orchestrate a microservices architecture critical for patient data management and real-time analytics. By adopting these tools, the healthcare organization achieved greater control over image creation and container lifecycle management. Buildah's flexibility enabled secure, repeatable image builds that met stringent data privacy standards, while Podman facilitated efficient deployment and scaling of containerized services across hybrid cloud environments. This approach led to improved system reliability, reduced downtime, and enhanced responsiveness to patient care demands.

Across these case studies, key takeaways emerge: containerization with Podman and Buildah on RHEL drives measurable performance improvements, including faster application delivery, enhanced security compliance, and operational efficiency. Organizations benefit from the modular, daemonless architecture that aligns well with DevOps methodologies and modern Infrastructure as Code principles. Furthermore, the ability to run rootless containers and integrate seamlessly with existing security policies ensures that container adoption does not come at the expense of safety.

These real-world examples underscore the value of embracing Podman and Buildah for container management, demonstrating how enterprises can harness the power of containerization on RHEL to foster innovation, resilience, and competitive advantage.

11. Future Trends in Container Management on RHEL

As container technology continues to evolve, Red Hat Enterprise Linux (RHEL) remains at the forefront of integrating cutting-edge advancements that drive enterprise-grade container management. Looking ahead, several key trends are shaping the future landscape of containers on RHEL.

Integration with Kubernetes and OpenShift is becoming increasingly seamless, enabling organizations to leverage powerful orchestration and management capabilities alongside Podman and Buildah. This synergy supports scalable, automated deployments and lifecycle management for containerized applications, while maintaining consistent security and compliance policies within enterprise environments.

Container standards and tooling are rapidly evolving to foster greater interoperability, flexibility, and efficiency. Emerging specifications such as the Open Container Initiative (OCI) continue to influence how container images and runtimes are built and managed, ensuring compatibility across diverse platforms and ecosystems. Enhanced tooling further simplifies container workflows, reducing complexity for developers and operators alike.

The rise of serverless containers and lightweight runtimes represents a shift towards more agile, resource-efficient deployments. These technologies reduce overhead by running ephemeral containers on-demand, enabling faster scaling and improved resource utilization. RHEL's ongoing support for these innovations positions it as a versatile platform for both traditional container workloads and next-generation serverless architectures.

Rootless container security and usability will see significant advancements, addressing usability challenges while strengthening isolation and protection mechanisms. Continued development in this area will make it easier for organizations to adopt rootless containers at scale, further minimizing security risks without sacrificing performance or developer productivity.

Together, these trends highlight a future where container management on RHEL is more integrated, secure, and adaptable—empowering enterprises to build, deploy, and manage containerized applications with greater confidence and agility in an ever-changing IT landscape.

12. Conclusion

Podman and Buildah have revolutionized container deployment and management on Red Hat Enterprise Linux (RHEL) by providing secure, flexible, and efficient tools tailored for modern enterprise needs. Their daemonless, rootless architectures combined with seamless integration into RHEL's robust security framework empower organizations to confidently build, run, and maintain containerized applications at

scale. These tools not only streamline workflows but also align with emerging DevOps and Infrastructure as Code practices, enabling faster innovation cycles while maintaining stringent compliance and operational control.

As enterprises continue to embrace digital transformation, adopting Podman and Buildah is a strategic move towards future-proofing application delivery pipelines and infrastructure management. Containerization is no longer just a trend—it is a foundational technology driving agility, scalability, and resilience in today's complex IT environments. Organizations that leverage these powerful tools on RHEL position themselves at the forefront of enterprise modernization, ready to meet evolving business challenges with confidence and efficiency.

References:

[1] Jena, Jyotirmay. (2020). Adapting to Remote Work: Emerging Cyber Risks and How to Safeguard Your Organization. 11. 1763-1773. 10.61841/turcomat.v11i1.15190.

[2] Mohan Babu, Talluri Durvasulu (2019). Navigating the World of Cloud Storage: AWS, Azure, and More. International Journal of Multidisciplinary Research in Science, Engineering and Technology 2 (8):1667-1673.

[3] Kotha, N. R. Network Segmentation as a Defense Mechanism for Securing Enterprise Networks. *Turkish Journal of Computer and Mathematics Education (TURCOMAT) ISSN*, *3048*, 4855.

[4] Sivasatyanarayanareddy, Munnangi (2021). Intelligent Automation in Action: Pega's Integration of AI and Next-Best-Action Decisioning. International Journal of Communication Networks and Information Security 13 (2):355-360.

[5] Kolla, S. (2020). Remote Access Solutions: Transforming IT for the Modern Workforce. International Journal of Innovative Research in Science, Engineering and Technology, 9(10), 9960-9967. https://www.ijirset.com/upload/2020/october/104_Remote.pdf

[6] Vangavolu, S. V. (2021). Continuous Integration and Deployment Strategies for MEAN Stack Applications. International Journal on Recent and Innovation Trends in Computing and Communication, 9(10), 53-57. https://ijritcc.org/index.php/ijritcc/article/view/11527/8841

[7] Goli, V. R. (2021). React Native evolution, native modules, and best practices. International Journal of Computer Engineering and Technology, 12(2), 73–85. https://doi.org/10.34218/IJCET_12_02_009

[8] Zohud, T., & Zein, S. (2021). Cross-platform mobile app development in industry: A multiple case-study. International Journal of Computing, 20(1), 46-54.

[9] Heitkötter, H., Hanschke, S., & Majchrzak, T. A. (2012, April). Evaluating cross-platform development approaches for mobile applications. In *International Conference on Web Information Systems and Technologies* (pp. 120-138). Berlin, Heidelberg: Springer Berlin Heidelberg.

[10] Amatya, S., & Kurti, A. (2014). Cross-platform mobile development: challenges and opportunities. *ICT Innovations 2013: ICT Innovations and Education*, 219-229.

[11] Majchrzak, T., & Grønli, T. M. (2017). Comprehensive analysis of innovative cross-platform app development frameworks.

[12] Biørn-Hansen, A., Grønli, T. M., Ghinea, G., & Alouneh, S. (2019). An empirical study of cross-platform mobile development in industry. *Wireless Communications and Mobile Computing*, *2019*(1), 5743892.

[13] Machireddy, J. R. (2021). Data-Driven Insights: Analyzing the Effects of Underutilized HRAs and HSAs on Healthcare Spending and Insurance Efficiency. *Journal of Bioinformatics and Artificial Intelligence*, *1*(1), 450-469.

[14] Dalal, K. R., & Rele, M. (2018, October). Cyber Security: Threat Detection Model based on Machine learning Algorithm. In *2018 3rd International Conference on Communication and Electronics Systems (ICCES)* (pp. 239-243). IEEE.