# Unix-Based Services in the Cloud

Harrison Smith
*Swinburne University of Technology*
*103068645@student.swin.edu.au*

*Abstract*—**This research report looks into the growing trend of integrating Unix-based network services including web, file, and database servers with the modern cloud computing platforms. The study aims to identify the advantages and disadvantages of such a process, while also touching upon the integration aspects such as deployment strategies, scalability, and monitoring solutions associated with these services. While the advantages such as enhanced security, cost-efficiency and scalability are significant, certain challenges such as complexity and the potential learning curve associated with Unix deployments and configuration is also discussed. The report concludes that despite these challenges, the benefits of integrating Unix-based network services into modern cloud platforms makes it a highly advantageous option for organizations aiming to update and improve their IT infrastructure.**

## I. Introduction

As the digital landscape evolves, integration of traditional systems with modernised cloud platforms has become a strong focus for many organizations aiming to achieve the pillars of operational efficiency and scalability. One particular integration that has amassed significant attention from enterprises is Unix-based network services with cloud computing platforms, with Linux now running 90% of public cloud workloads [1]. Unix and similarly Linux, have always been the backbone of enterprise computing and server infrastructure. Their modular architecture, robustness, extensive security options, and suite of networking tools make them ideal for cloud integration.

This report aims to explore the comparative advantages and disadvantages of using Unix on the cloud. The report will cover the integration of Unix-based network services, encompassing web servers, file servers, database servers on cloud platforms. It will delve into various aspects of cloud integration of these services including deployment strategies, scalability options, and monitoring solutions. Finally, the report will provide a comparative analysis of the basic tools available for managing Unix-based services in cloud environments.

## II. Unix on the Cloud

### A. Flexibility and Modularity

Modularity is particularly beneficial in cloud environments. Given the requirement for scalability and adaptability within the cloud, modularity is paramount to facilitate these requirements.

- **Advantage:** One of the reasons why enterprises have began to implement Unix-based systems is due to their modularity [2]. The operating system allows for easy customization and for scalability. It is both easy to add and to remove components without affecting the whole system. This makes it easier to adapt to these changing needs as previously discussed.
- **Disadvantage:** The flexibility of a Unix system leads to increased complexity. As such Unix systems often require specialized knowledge to configure when contrasted with other operating systems such as Windows. This can put pressure on organisations implementing these systems on the cloud.

### B. Security

With an ever growing cloud landscape, security is paramount.

- **Advantage:** Unix-based naively implement a strong security model including user permission, isolation of processes and a robust firewall. These features have been being extended to cloud services, providing an additional layer of security. This is particularly important in multi-tenant cloud environments where data isolation is so crucial. Furthermore, due to the open-source nature of Unix systems, the backbone of most security improvements is sourced from on demand community led patching efforts [1].
- **Disadvantage:** However, while offering improved security, these features also require careful management to maintain the security; as such a misconfigured system can be highly vulnerable and an entry point for attackers. Compared to other operating systems where security features are often pre-configured, it can sometimes be deemed more risk-averse to opt in to one of these alternative systems.

### C. Resource Efficiency and Cost Effectiveness

Resource management is a vital aspect of every cloud environment to ensure cost minimisation.

- **Advantage:** Unix systems tend to be less resource-intensive than other operating systems like Windows. This makes them ideal for cloud computing ecosystems where optimising resource usage results in significant cost savings for the organisation. Furthermore, many Unix-based systems are free or relatively inexpensive when compared to proprietary solutions such as Windows. This makes Unix-based systems a cost-effective choice for enterprises looking to move to the cloud without incurring high software costs.

- **Disadvantage:** Due to this resource minimisation, some distributions, particularly those used and implemented in cloud environments lack a graphical user interface (GUI), making them less user-friendly.

### D. *Scripting and Automation*

Scripting is a key feature used in cloud environments. This is often seen in across the entire development lifecycle and in security management.

- **Advantage:** The Unix shell provides users with powerful scripting capabilities making it easier to automate repetitive tasks. Automation in cloud computing allow for seamless management of large numbers of virtual machines or containers. Scripts can be used to automate cloud deployments, data backups, monitoring and logging, and more.
- **Disadvantage:** However, scripting can have a steep learning curve for those unfamiliar with shell commands, often incurring costs associated with upskilling.

### E. *Compatibility*

Cloud environments are often operated in a hybrid manner, in which some infrastructure is hosted on the cloud, while some is physically on-premesis.

- **Advantage:** Unix-based systems are regularly more compatible with types of hardware. As such they can be easily integrated with other systems and applications, particularly within these hybrid cloud environments where systems must work seamlessly together.
- **Disadvantage:** While these systems are often compatible, specialised hardware can have limited driver support.

## III. Integrating Unix-Based Services on the Cloud

### A. *Web Servers*

A web server is a software application which processes, stores and serves web pages over the internet. A Unix-based web server is one that runs on a Unix-based operating system such as Linux or macOS. Unix-based web servers are renowned for their security, performance and their modularity. Linux infrastructure tends to offer 20% more throughput than it's OS counterparts [4]. As such Unix-based web servers are a commonly used within cloud ecosystems, allowing for customization, scalability and performance.

*1) Deployment:* Within a cloud environment, web servers are regularly hosted on a Unix-based virtual machine (Azure) or an EC2 instance (AWS). On top of this a web server software will be installed and then configured accordingly. Unix-based containerisation technologies are often implemented on the virtual machine to encapsulate the server software and its dependencies. When deploying virtual machine/instances, cost is a vital factor. [4] Leoni identified that the premium for running a windows instance over a Unix-based instance starts at a minimum of 16% with Google Cloud and increases to 27% with Microsoft Azure.

*2) Scalability:* The modular architecture of Unix allows for easy scaling, particularly horizontal scaling. This is where additional web server instances can be added behind a load balancer. Horizontal scaling is the preferred method for scaling in the cloud to prevent resource deficits [3]. Unix's stateless design is crucial for horizontal scaling, allowing requests to be distributed without requiring shared state. Unix-based web servers often operate in this stateless manner, implementing techniques such as token-based authentication to maintain a stateless operation in those with dynamic content. Alternatively, vertical scaling, which is much less common in cloud settings, could also be facilitated by the ability to add resources to a single instance.

*3) Monitoring:* Cloud service providers offer their own solutions for monitoring, which allow for real-time analytics of performance tracking for web server instances. However, if instead opting for an open-source alternative, there are multiple tools such as Prometheus widely used with Unix-based systems, which collect configured metrics on targets and provide alerts.

### B. *File Servers*

A file server is a server which provides a centralised storage location, allowing the management and access of files and directories over a network. Often implemented in organisations to allow for data sharing and remote access of data. Unix-based file servers are known for their security, robustness and ultimately their performance over the alternatives. They often use the NFS (Network File System) protocol [6], which is native to Unix systems for file sharing. These are well suited for cloud environments due to their scalability and cost efficiency. Amazon EFS, Microsoft Azure File and Google Cloud Filestore all fully support NFS [6].

*1) Deployment:* Each cloud provided offers its own file storage service. Some of these file storage services such as AWS EFS are exclusive to Linux based systems. An organisation can opt in to use a provided Unix-based storage service or they can manually deploy a virtual machine with a Unix-based operating system to act as the file server. If manually deploying such a server on the cloud, it must be carefully configured. The NFS server should be configured and started, including setting up permission, exports and firewall rules for access control. Furthermore security measures such as IAM (Identity and Access Management) should be deployed on top on the machine to control who can access the server.

*2) Scalability:* Scaling a manually created NFS file server on the cloud can be achieved in several ways. As previously discussed, it is often recommended follow a horizontal scaling method, in which the load is distributed by adding more virtual machines. This can be implemented to auto-scale in which the server begins to scale as load increases and decreases. Each cloud providers offers a load balancing service, which should be used alongside this to distribute file requests across multiple file server instances. Furthermore, for larger data sets, data can be partitioned across multiple file servers or even geographical regions to allow for faster retrieval and logical segregation.

*3) Monitoring:* Monitoring Unix-based file servers in the cloud is crucial to obtain insights into security, system health and performance. This can assist in making informed decisions. Similar to web servers, cloud-native monitoring tools are often implemented to monitor system usage and performance. However, for more granular events, log management is vital. There are some open-source solutions [7] which can streamline this monitoring process by aggregating and analysing logs. These tools should be connected to NFS logs, file access logs and syslog.

### *C. Database Servers*

A database server is a server that hosts a (DBMS) database management system, and facilitates storage, retrieval, and management of the data in databases. A database server serves as the backend to many applications and services, handling read or write requests on data. Unix-based database servers are so popular due to their high performance and low costs, which is a crucial feature. A Windows Server 2016 Datacenter starts at $6,155 [8] while Linux can be obtained with no license fee at all.

*1) Deployment:* Database servers can either be deployed using a managed service offered by cloud providers or by manually developing and securing one on a virtual machine. Deploying a database server on a virtual machine involves choosing a Unix-based operating system for a virtual machine, and then installing the required software. Using a package manager, an open-source software can be installed and configured. Alternatively, managed database services offer many Unix-based [5] database engines. The managed services can be configured to handle routine database tasks, while serving the database on a Unix-based operating system.

*2) Monitoring:* Similarly to other Unix-based networking services on the cloud, monitoring can be performed through monitoring tools, as well as, if implemented through a managed service, through specific metrics for that service. The primary metrics to monitor include server performance and utilisation as well as database connections for security purposes.

## IV. Tools for Managing Unix-based Networking Services in the Cloud

*1) Docker:* Docker is a platform that uses containerisation to package applications and their dependencies together. This streamlines to deployment and allows applications to run consistently across various environments. Docker is commonly used to deploy Unix-based web servers, databases and other network services within organisations. Considering cloud environments might involve a mix of different services and infrastructure, a tool such as Docker is particularly useful.

**Pros:**

- Portability: With Docker, containers can run anywhere. This ensures consistency across different stages of development and deployment.
- Resource Efficiency: Unlike virtual machines, Docker virtualises the application, not the OS. Docker containers share the host system's OS kernel. This makes them lighter and faster than traditional virtual machines.
- Isolation: Each Docker container is isolated from one another. As such, this improves security by containing any potential application breaches to the individual container.

**Cons:**

- Learning Curve: Conversely, Docker, as it is its own software involves a different approach to configuration and requires users to understand and learn new commands and concepts.
- Complexity: For larger applications such as a complex Unix-based database server which interacts with a Unix-based web server, orchestration can become challenging. Each service requires its own container, configuration and integration.

*2) Ansible:* Ansible is an automation tool used for configuration manage and deployment. Ansible to designed for Unix-based systems, and as such it is often used to automate the setup and maintenance of these Unix-based web servers, file servers and database servers. This tool allows for a seamless and efficient deployment.

**Pros:**

- Ease of Use: Ansible uses YAML (Yet Another Markup Language) for configuration scripts, also known as ”playbooks”. YAML is a simple and easy to read language, ensuring a smaller learning curve for developers creating configuration and automation tasks.
- Agentless: Ansible runs in an agentless manner. This means that the software agent does not need to be installed on the target server. Ultimately, this reduces overhead, improves performance and simplifies the setup process.
- Extensible: This tool is designed to be modular. Users can write their own modules to perform certain tasks. As such this tool can be tailored to a niche environment as most Unix-based environments are.

**Cons:**

- Performance: Unfortunately, since Ansible is agentless and instead operates over SSH, each task must initiate a new SSH connection to the server. This can lead to slower execution times.

*3) ELK Stack (Elasticsearch, Logstash, Kibana):* The ELK stack is a collection of three open-source tools that work together, allowing you to search, vizualise and analyse data in real time. Specifically, Elasticsearch provides a search engine, Logstash is a pipeline for data processing and Kirbana is a data visualisation tool. The stack is often used for tracking and analysing log and event data as well as monitoring servers in real time.

**Pros:**

- Scalability: Alongside Unix-based server deployments, the ELK stack can scale horizontally to handling increasing load. This makes it suitable for high volume enterprise environments.

- Visual and Real Time Analysis: The real time and visual nature of the stack allows for effective monitoring of server logs and security event analysis.

**Cons:**

- Complexity and Cost: The stack can be difficult to set up and manage, particularly for those who have little experience. Additionally, depending on the size of the application/load of the servers, the larger scale deployments can become costly.

*4) Prometheus:* Prometheus is a open-source monitoring and alerting tool designed for scalability. It is very suitable for supporting Unix environments due to it's ability to monitor complex and dynamic environments.

**Pros:**

- Scalability: Prometheus is well suited for cloud applications, integrating with dynamically scheduled services. In a complex Unix-based network environment, such a tool can handle multiple monitored targets.
- Granular Monitoring: Prometheus supports a variety of hardware, kernel, and other OS related metrics. When correctly configured, this allows users to obtain a granular view of the servers status at a glance.

**Cons:**

- Complexity: Prometheus implements its own language 'PromQL'. Whilst being powerful in itself, it introduces complexity for newer users when attempting to use the tooling to its full capability.

## V. Conclusion

Throughout this report we have explored the various advantages and disadvantages of Unix-based service integration within cloud environments, touching upon critical the aspects of deployment, scalability, and monitoring.

Our analysis reveals that Unix-based systems provide a large range of benefits when deployed in cloud environments, including enhanced security, cost efficiency, improved performance, and most importantly, greatly improved modularity and scalability. However, these advantages do come with their own set of difficulties including the complexity of implementation, learning curve associated with maintenance and required learning for effective monitoring via common tools.

As cloud computing continues to develop, the tools, methodologies and support for integrating Unix-based network services are likely to improve, eliminating some of the current posing challenges. Ultimately, despite the difficulties discovered associated with complexity, it was identified that the increasing support for Unix services in cloud environments is due to the undeniable organisational cost and performance improvements.

## References

[1] *(1) Role of Linux in Cloud Computing — LinkedIn*. URL: https://www.linkedin.com/pulse/role-linux-cloud-computing-manjeet-singh/ (visited on 09/06/2023).

[2] *Benefits of Using Cloud Based Linux Systems in the Modern Enterprise*. Alibaba Cloud Community. URL: https://www.alibabacloud.com/blog/benefits-of-using-cloud-based-linux-systems-in-the-modern-enterprise_596497 (visited on 09/06/2023).

[3] *Horizontal Vs. Vertical Scaling: Which Is Right For Your App?* URL: https://www.missioncloud.com/blog/horizontal-vs-vertical-scaling-which-is-right-for-your-app (visited on 09/07/2023).

[4] Ken Leoni. *Linux vs Windows – a cost comparison*. Longitude. May 16, 2018. URL: https://www.heroix.com/blog/linux-vs-windows-a-cost-comparison/ (visited on 09/07/2023).

[5] *Unix Installation*. PostgreSQL Documentation. Jan. 1, 2012. URL: https://www.postgresql.org/docs/6.4/odbc18456.htm (visited on 09/07/2023).

[6] *What is NFS Shared Storage in the Cloud? — Synopsys Cloud*. URL: https://www.synopsys.com/cloud/insights/what-is-nfs-shared-storage.html (visited on 09/07/2023).

[7] *What is the ELK stack? - Elasticsearch, Logstash, Kibana Stack Explained - AWS*. Amazon Web Services, Inc. URL: https://aws.amazon.com/what-is/elk-stack/ (visited on 09/07/2023).

[8] *Windows Server 2022 Licensing & Pricing — Microsoft*. URL: https://www.microsoft.com/en-us/windows-server/pricing (visited on 09/07/2023).