National College of Ireland

Multi-Region Docker Container Connection: A Comprehensive Broach Using Network Functions and RDS Cloning

MSc Research Project
Programme Name

Jayesh Chitnawis
Student ID: x22244328

School of Computing
National College of Ireland

Supervisor: Rashid Mijumbi

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Jayesh Chitnawis
Student ID:	x22244328
Programme:	Programme Name
Year:	2024
Module:	MSc Research Project
Supervisor:	Rashid Mijumbi
Submission Due Date:	16/09/2024
Project Title:	Multi-Region Docker Container Connection: A Comprehensive Broach Using Network Functions and RDS Cloning
Word Count:	9543
Page Count:	26

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Jayesh Chitnawis
Date:	16th September 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Multi-Region Docker Container Connection: A Comprehensive Broach Using Network Functions and RDS Cloning

Jayesh Chitnawis
x22244328

Abstract

This research will explore the incorporation of network functions virtualization with the combination Virtual extensible LAN (VxLAN) and OpenVSwitch (OVS). This amalgamation of networking technologies is used to overcome key limitations in traditional network setup which are responsible for network latency, scalability and data availability. This study has proposed a solution to enhance network infrastructure for containerized platforms. Here significant focus is assigned on data availability as which cannot be carried by network functions so we implement RDS multi-region and Multi-AZ's. Findings will exhibit this infrastructure proves significant improvement in network performance, and also major constraint confirms that one change performed in one node is accurately reflected across the nodes if required. This simulation is performed with AWS environment to create hybrid cloud environment. This implications of study are significant and routed pathway as it provides resilient nature and advanced features as well. This commercialization is also considerable we have developed managed services that combine and provides enterprise operation in hybrid as well as multi cloud environment

1 Introduction

Endeavouring and Navigating in the landscape of IT there are certain swift advancements in cloud computing and technologies. Over these periods it took the revolutionized form in communication, technology and in every aspect of another field. This IT platform provides us with the platform for enabling big or small organizations to deploy, scale, manage and maintain applications with matchless capabilities. This transformation has reached high limits now. But if someone wants to grow then the sky is the limit. As in this field advancement is limitless there will be no end. All these high achievements began with the start of communication and even at this time, the IT industry of the cooperate world is at the core of traditional methods which are particularly used in Docker. Docker allows applications to be packaged and functions smoothly on any platform deployed. In modern architecture docker containers and their efficiency have become the cornerstone due to their inclusive nature as lightweight and suitable for all environments.

The most trans-formative milestone in IT was the invention of **Virtualization**. The usage of computer resources was changed, it was revolutionized resources were shared as efficiency, and scalability increased up to high standards. No doubt it was an advent in Industry. Even today, we rely on Virtualization for cloud and cloud applications. As the

cloud has grounds for virtualization. It has become a backbone for Industry, prevailing unparalleled flexibility and high scalability in managing IT infrastructure. Virtual machines in virtualization virtualize the whole operating system, whereas containers have a lightweight approach to operating applications. As a result, docker became **leading platform** for containerization. Docker deployment has proved to be streamlined in application deployment. These contain adapt in size and also compatibility maintaining the containing network becomes increasingly complex. Humans do try to get a grip on necessitating robust solutions as it will confirm efficient communication not only within an infrastructure but also with **manifolded distributed networks**.

A diverse network brings up a diverse and vivid environment. There are numerous challenges which we face with containers.

1. **Network latency and bandwidth limitations.**

    - Problem: Industries are spread across the globe in different geographic locations and when communication is provided via containers which are spread in multiple networks are considerable concern. Some nodes are distant here data packets take a long time to travel, which eventually leads to delays. Here we can consider bandwidth can become throttle for data transferring rates.
    - Bandwidth in the network is fixed in prerequisites if latency is high and we have limited bandwidth this aspect can be the reason for degraded application performance. This majorly impacts real-time applications as they require high throughput. Raghunathan (2021)

2. **Security Aspect**

    - Problem: Multiple distributed network involves multiple entry points, which leads to an attack surface, secure communication is a necessity here. A network must have robust encryption, and some authentication factors and mainly network isolation in containers is vital as it contains information.
    - If there is no proper isolation in the network especially in a containerized environment. There is a high chance of data interception or tampering of data. This sums up to loss of sensitive data.

3. **Consistency and Coordination**

    - Problem: Any alternations in one node must be mirrored in other nodes as well. And that too quickly and spontaneously and most importantly accurately this ensures consistency. Maintaining this level of synchronization is challenging. Simeone et al. (2008)
    - This can lead to inconsistent data and errors. This results in improper service of an application and users may face different performances in different networks or regions.

4. **Scalability Matters**

- Problem: As the number of industries is growing, the population is growing hence we need more services to accommodate all of our clients. More applications are required to serve more people so we need more containers as well. Scaling network according to containers is quite difficult.
- This scalability issue can become a significant reason for bottlenecks where the network will not be able to keep pace or speed with increasing demand. This may also suffer to recognise areas where demand is reduced, hence reduction in utilities will not be sufficient.

5. **Interaction and Congruity**

- Problem: Maintaining seamless interoperability is quite challenging as the globe consists of numerous networks and every network has its own set of protocols and standards. This is an ongoing problem in multi-cloud as well because providers have to provide their infrastructure.
- This leads to a lack of compatibility and interoperability in complex networks which may also result in failed communication. Some specific solutions are necessary to bridge different systems and environments.

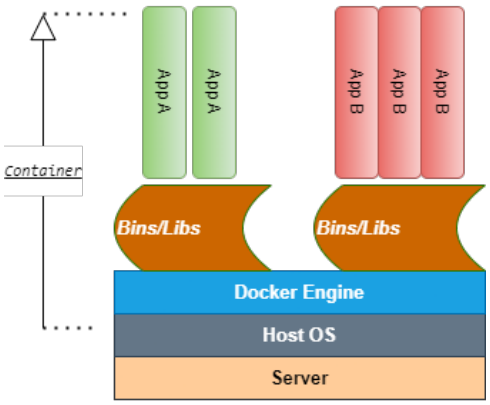

Figure 1: Container Technology
Liu and Zhao (2014)

To store and distribute container image docker replies on image registries. In the docker image, which consists of a registry here each layer is stored as a tarball and identified based on their content. It uses object technique to store or retrieve layers. A large-scale-object setup stores each layer as an object in the Large-scale-object store. Some conservative estimates show that in spring 2019, Docker Hub alone stored at least 2 million public images totalling roughly 1 PB in size Nannan Zhao and Butt (2020). But we came to a point where the demand for storage is getting worse because data is too huge and getting duplicated again and again. It is a replica from one server to another server of the database by our research we ensure that these replicated databases are synchronised and also serve for fault tolerance and load balancers.

For shipping and running applications within containers and deploying them on desired platforms docker is one of the best platforms. As replicated databases increase

within distributed systems availability and flexibility of applications increases. Providing a robust environment for deploying and managing these replicas.

Addressing these researchers Software-defined researchers have come up with advanced networking technologies and some special features like OpenVswitch (OVS) and the extension of LAN in a Virtual environment that is VxLAN. There are other technologies developed over a period as well. Software-defined network, network function virtualization (NFV), edge computing and many more. These techniques have proven track of overcoming -traditional problems with significance. In terms of flexibility or scalability. Here multi-cloud environment is based on diverse networking, as cloud networking plays a basic role in communication and other purposes. Why not use a combination of these techniques and make better and more reliable infrastructure for docker as well?

In current years, the integration of VxLAN and OVS has replaced many issues in technical and network infrastructure as proven. Researchers have built our prototype to deploy clusters using docker containers and interconnecting the host using VxLAN tunnels.Babu (2016) An interesting feature of OVS is that it is a flexible and controlling programming interface that can efficiently manage network traffic and here it will be a virtualized environment with VxLAN. This enables us the create of overlay network. This facilitates this large scale infrastructures hence we can use it for containers as well which will initiate us to establish communication across multiple networks.

Technologies have shown such significant updates in an ecosystem and maintaining and managing the whole concept is improvised. Yet in docker, there are some aspects which need to get some attention as they exist but they could be improved for a better experience. Network function Virtualization is a cutting-edge solution here. A prominent way for decoupling network functions for every issue as if traffic is getting more then we can apply routers as a function, if the security aspect is a concern then we can use firewalls, if traffic is uncontrolled then go for load balancers. NFV enable us to set these functions as software. This radical shift does not just make convenience for deployment and efficiency but it does have a proven track for scalability and proficiency in networking, NFV orchestration did take responsibility for the global management of network resources and network services also referring T-NOVA project.Pattaranantakul et al. (2016)

Our thesis also aims to demonstrate the combination of these networking techniques to overcome traditional limitations. And provides a resilient network infrastructure despite VxLAN and OVS for communication and provides additional features by NFV functions integration. In this research, there is another key aspect as well which involves mimicking, the cloning the utility of Relational Database Service (RDS) multi-region and multi-availability zone in nodes. This cloning function is made to replicate, shared capabilities for cloud services which will result in high throughput in docker and automated backup as well. This all over resulted in seamless fail-over. By the success of this research, we can prove a seamless robust, fault-tolerance database. This well led to a high consistency rate. And supports regional outages and will not fall for network failure. Overall this experiment will integrate in architecture strength.

This technique will redefine the most possibilities within the Industry for communications as well as data transfer. This network infrastructure will archive results by pushing the boundaries of virtualization. You will get to know the resilient structure of docker in analysis.

1.1 Research Question

Docker is quite a popular technique in the current world fulfilling the required demands and running swiftly in modern updates. Some significant challenges are experienced when clusters are formed together and situations are transferred across other networks. To address these critical confrontations, we are implementing OVS in VxLAN with virtualized functions. Here we get a solution for cross-connectivity, but as it is across multiple regions and networks, we can face situations such as data packet loss. Additionally, we provide high data availability through mimicking RDS across network cloning multi-region and multiple- AZ replication plants

The primary question in this research focuses on answering: **"Can integration of Network Function Virtualization in network infrastructure combined with RDS cloning and mimicking techniques, eliminate limitations of VxLAN setup and data packet loss when deploying clusters of containers of Docker in multiple networks and regions or environments? "**

2 Related Work

2.1 Network Functions Virtualization (NFV) and 5G technologies

Network Repository function have ability to run in Docker while deployed on nodes, is checked here. This results in the docker based network functions platform can hold 5G technology, this technique is capable of holding 5G network. Which leads to crucial for upcoming generation. Zeng et al. (2017) Network functions can run as software on a VMWare or a Virtualized Hard Disk. Hence this is quite in demand as it provides greater flexibility. This is required for the future generation high traffic. This paper also talks about 5G architecture and used of generation partnership project which are based on clous native architecture. This all is achieved by NFV technology which makes modernizing telecommunications networking applied knowledge used in day-to-day work rather than old traditional work and remove plugins dependencies. Somewhere this author also writes about European telecommunications Standard Institute, which been a vital role. As docker is also based on Linux. Hence it is ideal for implementation of NFV. Which makes it suitable for the experiment for the dynamic and resource intensive needs. Hence here author focuses on container communication and network isolation which refers to security concern. Network within containers is quite important to be carried out. This paper have evaluated the performance of mainstream platforms which are flannel, docker swarm, and calcio. Liu et al. (2019) He talks about network model which refers to the docker libnetwork project, it acts as standardize interference between components as docker daemons and some of the network drivers. Sandbox was also a part of it. The coreOS and CNI defines lays a easy contract between container and plugins in runtime. Facilitating the charge and discharge of network interfaces here they have used JSON scripts. Hence provide utilization in wide range. As per solution it also requires containers network solutions flannel that is actually designed for Kubernetes lightweight is satisfies the markings and easy to configure as even same goes for calcio and provide high flexibility with advantage of multi subnet isolation mechanisms.Zeng et al. (2017) Load balancing handles by their internal functions. Here one drawback was highlighted as itself it suffers from performance lose and reason for it is encapsulation with decapsulation process.

Here researchers have used big data sets from Zhang, Zhang, Wu, Lu, Wang and Mao (2020)Kaggle some of them contains 129463 dockerfiles. As it is dataset, here it acts as dataset , it goes under filter validation process here we will talk about containers pairs and construct the overall dependencies in network. The experiment is actually performed for source containers and target containers from the Docker files. In results researcher have achieved the outcome for dependency network comprises 83030 nodes and 98428 edges. Here we can say that loose interconnection compared to other world real networks. In the diameter of 23 and shortest path of 5.72 some apes containers identified which includes Ubuntu, Alpine, Node and Python there are more vital for ecosystem. Sub network analysis categorized in core containers are in 5 different forms Operating System(OS), Language Runtime(LR), Tools or Services(TS), Application Framework(AF). Here OS sub networks shows the higher interconnections, Indicating frequent between nodes which shoe case dependencies. Here analysis of more than 120,000 docker flies gave the robust basis for conclusions hence data processing network construction ensure reproducible.

Network architecture have seen significant improvement with NFV and also with SDN, their integration is also one powerful impact tool. This literature will show us the key contribution for security mechanisms through VNF's implementation. Network function virtualization have ability replaced specialized network, as it deploys virtual function.Chin et al. (2023) And perform the tasks which was performed by hardware, so here this hardware is replaced and it becomes easy to manage virtual functions and also get resources utilization in adequate manner. Hence, we achieve results like decoupling of network, is the reason we perform network isolation. But in IOT when we integrate NFV/SDN, we get lack of adequate security capabilities due to their limited resources sharing ability. For the reason there we developed cryptosystems within NFVI, so here VNFs are virtualized and delivered as crypto engines. And use case it only for IOT devices. This is incredible development innovated to secure NFV/SDN, IoT devices. Even throughputs are in range of 603.78 Gbps around.

Traditional hardware-based function or network appliances are replaced by NFV, because it's been a transformative technology in every aspect like telecommunication. Cai et al. (2023) This paper will guide us with overall evolution of NFV over decades. It was used first conceptualized in 2012 as means to decouple network. This was a significant marked for traditional approach too network management as they were dependent on more hardware resources. Industry specification group has played a important role for standardizing NFV and its integration across multiple vendors, also an vital role in widespread adaptation in 4G and 5G. These are remarkable development and these developments are not possible without challenges. Major concern in NFV was for fragmentation. And everyone operated from all different areas and regions so no proper unified standards. And high time in areas on containerization. They do require careful considerations. But this integration has provided us autonomous networking and automation. Cai et al. (2023)

Service function chaining (SFC) is a new concept in modern networking. This topic is similar to SDN and NFV. As names says, it is a chain of network services like load balancers, firewall and intrusion detector. So, traffic goes through this system of layers. At this junction SFC is paramount. SFC is also one reason that traditional static and network rigid hardware architecture are replaced to virtualization functions allowing them to run and serve general purpose. This is transition from static to dynamic. Kaur et al. (2020) Heterogeneous network needs better integration for SFC, as this was challenging phase varying in service requirements. As we need more intricate architecture here. SFC

is also been provisioning SDN and NFV. Offering great flexibility and reliability.

2.2 Software Defined Networking and Multi Controller Architecture across Domains

Paper contains come related work which involves mitigating single-point failure in SDN and also include distributed SDN which are OpendayLight or some hierarchical SDN. We can also some adaptive multi-controller architectures are also mentioned in previous work. As the matter of fact, it can't be denied that they do includes come complexities in installation and require high maintenance. Software defined networking is standard approach for networking in which controllers are used, that are based on Software of an API for directing Traffic on a network or applying a hardware infrastructure. It can also decouple control plane from data plane and provide a usefulness by centralized controller. This paper gives a clear vision on bottleneck which is experienced by single centralized controller as a single point of failure for the growth of network. Whereas in this research we can see that author has proposed a multi-controller system which have ability to resolve this issue, and provide some perks like availability and reliable over services. Here author have proposed a new way to provide deliver with docker swarm. They state it as "cooperative scheme" for combining multi-controller SDN along with Docker swarm, here docker swarm will manage clusters which will act as single server and hence it can facility orchestration and clustering advantages. As result it was observed high availability in system and better fault tolerance.Babayigit and Abubaker (2023)

Paper has addressees some challenges which affects the deploying network services efficiently across software defined network and they also shown the integration of network function virtualization. Here paper also talks about power consumption while load balancing and scaling. While the previous reported in paper have some limitations which are overcome in this research as related to dependencies error and insufficient service provisioning so to make these limitations turn into advantages. author used NFV and allow functions to run virtually which made it more effective than any other traditional system. Zhang, Wang, Dong, Zhao, He and Huang (2020)

System Framework.

Here are some discoveries which are quite related to our thesis topic.

- Topology Discovery: maintain network topology on global view.
- Energy Detection: servers energy consumptions
- Network Monitoring: Tracks networks status and also the impact of server workload.
- Resource: server have resources which are collecting information and hopefully having it in data form.
- Knowledge server: SFC deployment requires various data that has been stored.

Load balancing is crucial in any environment and to handle traffic on client side it is important, can't afford mistake in it. Here VNF's was introduced to increase efficiency in load balancing, many strategies are provided in paper as result is also consist of contribution by different scientists

2.3 Docker Swarm- based SDN multi Controller Architecture for Enterprise Networks

This is a popular java script framework actually it simplified the packaging and distribution of web applications in docker. Here it ensures consistent runtime environment, facilitating the angular application with deployment of angular. This method has actually enhanced application performance and scalability. Being concern about security. Here security concerns are crucial as this is one thing which cannot be eliminated nor left unattended, here studies have various security aspects which include vulnerability assessments, and container isolation which results in securing image building practices. This is why security is essential in docker images which leads in protecting applications and keep them away from potential threats.Badisa et al. (2023) Hence this review addresses security consideration and performance evaluation methods which will affect comprehensive summary which required in current docker applications. Such techniques are called "being lazy couch potatoes" which refers to eliminating unnecessary dependencies did not actually prove to be effective. Optimizing parent size image was quite effective as reduction achieved was 56.32 percentage. The important one integration of Nginx was resulted in to 97.17 percentage , this was significantly improving deployment efficiency.

My proposed implementation can relate some common points with this ideology. We both address the issues of docker and its ability to manage containers hence this also prove package applications and their respective dependencies and available through the outbreak of containers. Muzumdar et al. (2024)Hence this nature makes it consistent across different environment. Where one organization can deploy or make it available anywhere without concerning of dependencies and being relax about network. As it can survive in different network.

2.4 Orchestrating Micro-services beyond Containers

This research have explored the field of orchestration of microservices on platforms like Minikube, Kubernetes, Docker and Docker compose. Here experiments has been carried out which have enabled seamless deployment and scalability of microservices architecture. Modern software tools are utilized for highlighting their ability to enhance development workflows and also the deployment with optimum use of resources. Eyvazov et al. (2024)

Orchestration tools and their integration

- Minikube: base for Kubernetes environment locally, testing and development.
- Kubernetes: containerized applications are managed and deployed using Kubernetes.
- Docker and Docker Compose: making application containerized and managing multi containers.

Here the comparative analysis for respective techniques which results showed Kubernetes in quite high standards from anyone else, due to its advanced features and scalability. Whereas docker compose is suitable for local development and Minikube do have some moderate features. Some advantages of microservices in Kubernetes as it the automated health checks and importantly self healing ability. As this conditions are ideal for managing micro services architecture.

Security challenges with docker containers image within cloud. Researchers have proposed here a vulnerability centric approach for this branch and identifying and mitigating vulnerabilities in images. Some of the use cases are provided in paper eg. Ahamed et al. (2021) NIST SP 800-190 some security guides are aligned with OWASP Container Security Verification Standard. We can see through the experiment that author tries to enhance security through audits, and ensuring compliance. Docker images have revealed some common vulnerabilities which similar to package versions, root privileges and orchestration and configuration. We have to focus on securing images which will prevent it from DoS or MitM attacks. Multiphase methodology has proven through test and mitigation techniques for docker image. NIST and OWASP standards are important as they ensures the robustness and relevance for security measures. OWASP CSVS are covering security requirements across 3 levels L1,L2,L3. It will meet stringent security standards and also mitigating potential risks during life cycle.

3 Research Niche

Currently industry have large span of cloud driven vehicle. As demand for resources and Utilization also increases. To scale this demand, high availability and fault tolerance in distributed systems which supports hybrid environments is greater than ever. Industry relies on container-based applications and database on cloud will provide, seamless application across over the globe. To solve these predicaments researcher have performed some experiments over aspects in container platforms for orchestration, network performance, networking metrices, latency etc. here it is one aspect to manage inter-container communication and data consistency if, it is travel through multiple regions. As now applications are deployed numerous geographical reasons, here there is necessity of robust and resilient architecture while prove data availability and consistency and also minimize latency.

4 Methodology

4.1 Overview

Our research methodology aims to enhance scalability in Docker networking and its compatibility with different network regions. With the integration of NFV, we can achieve significant performance of Docker container clusters. NFV-enabled docker containers and their deployment and management across different regions with multiple networks. Cloning the RDS MySQL database process supports the main aim of cloud services. That is consistency at every stage and at every node and automated backup and accessible to every user in demand and this system has proved to be less network failure.

This section will outline the systematic approach and process nearly new in conducting the research for enhancing docker deployment and wearing the new network infrastructure using OVS, VxLAN and NFV. The methodology will provide a clear idea for our experiment setup and structured process. Precise steps for network infrastructure and replication of RDS at nodes. And some evolution-concentrated approaches towards system performance.

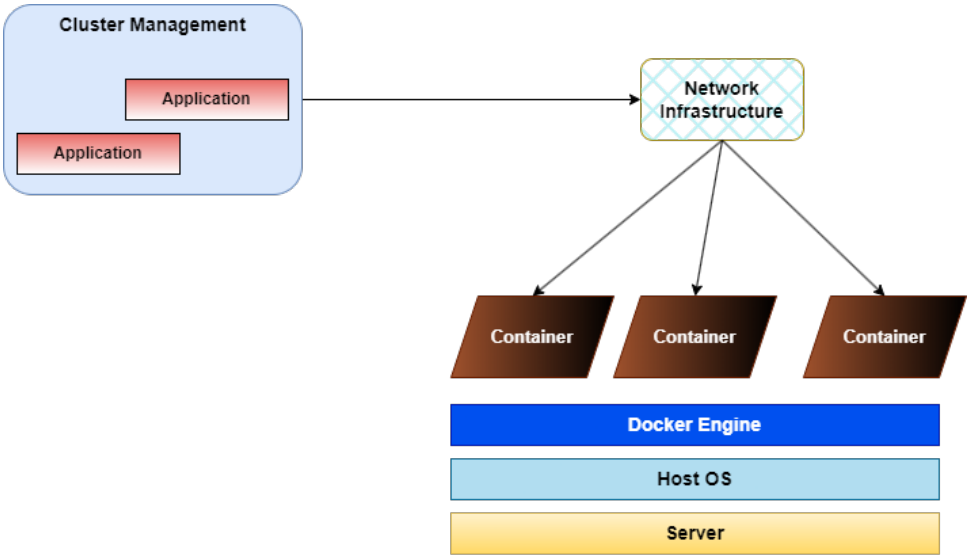

Figure 2: The architecture of platform

4.2 Research procedure.

This research will lead to a systematic approach to robustness in infrastructure for docker containers. Provided an architecture for containers which have master nodes and slave nodes. Which works on the principle of push architecture. This structure will facilitate the management of a database or an application deployment and will deal with the demand for containers as they need to increase or decrease across regions with diverse networks.

A resilient overlay network infrastructure based on VxLAN and OVS. Database across distributed environments requires dynamic and incapable of failing while communicating. Not to mention, with high data availability and consistency in RDS we can achieve mimicking multi-region and multi-AZ, here we ensure any changes in the master node are reflected across slaves nodes which accepts the primary node as the master node. Here data maintaining consistency is the overall enhancement of reliability in a system

4.3 Design and Setup

- Docker Nodes: it consists of Master node and Slave nodes. Here Master node is responsible for controlling and managing the deployment process within slave nodes. Our master node is equipped with Docker swarm in orchestration for executing whereas Slave nodes are responsible for executing commands transmitted by the master node and provide the work and status of the work.Pattaranantakul et al. (2016)

- Push Model. The important reason to choose the push model here is, that any updates or changes are responded to all nodes immediately so that is how we maintain consistency. As a result, push model architecture is crucial to maintain consistency. So all nodes are operated and they receive updates changes and provide the latest latency.

4.4 Network Infrastructure Configurations

- OpenvSwitch (OVS): OVS is installed and configured at each and every node. The main aim of providing OVS is to control traffic and manage, load in the network in efficient way. Some features provided by OVS are control flow, network segmentation and Packet filtering. As a result, it does have a proven track, it was eligible to manage complications in traffic and provide us with the best results.

- VxLAN Setup: it will be the base of the Overlay network. We can achieve over goal with the help of an overlay network as it will allow containers to communicate across a variety of networks. And we took the help of Virtual tunnelling endpoints (VTEPs). As they connected on every node this allowed us to perform operations like encapsulating and decapsulating the reason they traverse in network. With the extension of Layer 3, we have achieved the flexibility that was needed across applications to manage themselves. And also, in different regions and vivid data centres. Here VxLAN connects one environment to another environment, all traffic is routed from NFV as Virtual router Network functions are installed as it will route all data packets first and provide more security in infrastructure.

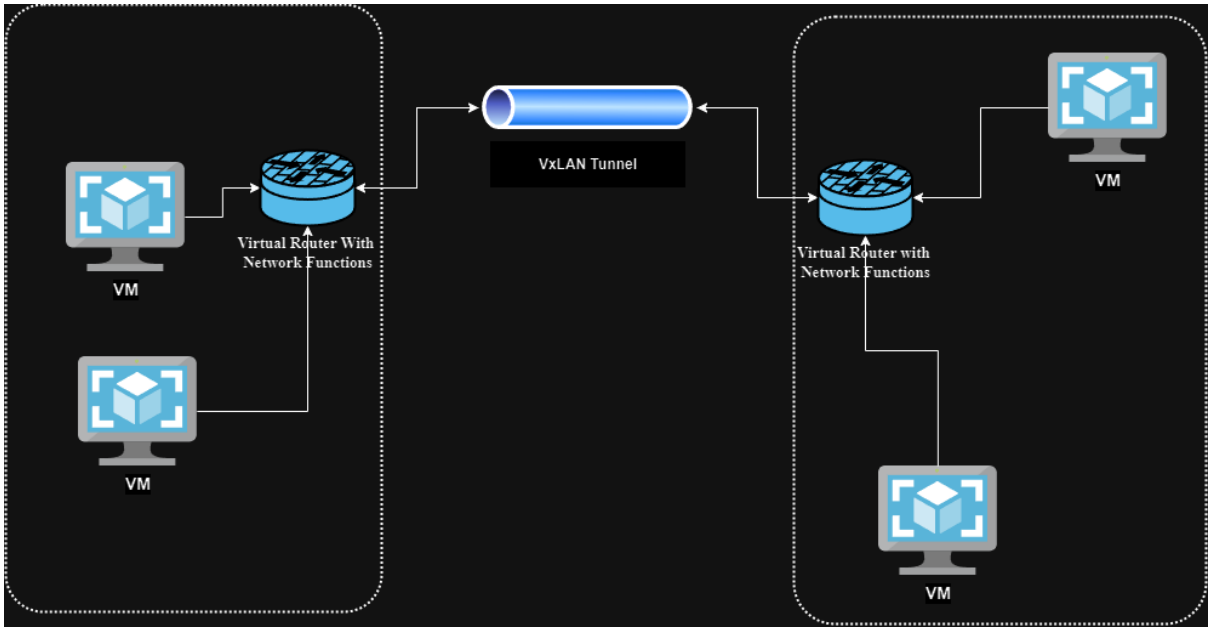

Figure 3: VxLAN setup / Overlay Network

4.5 Equipment

For demonstrations we out setup is currently established on private cloud. And consist of some virtual machines which will be useful as they will be play a part as real data center of different network regions. This simulation will consist of following parameters:

- AWS EC2: as we can demonstrate our docker in EC2 and make their nodes available on other region EC2 instances as well.

- VMware Workstation: creation and deployment of local VM's they will act as another region and here and on EC2 we can add additional slave nodes as we represent a hybrid environment.

4.6 Data collection and Analysis

We have recorded data through system monitoring and considering previous work where similar environments were createdBabu (2016). Software integration and network analysis with virtual environment.

- Network traffic analysis: we have used Wireshark to analyse and capture the logs of data with network traffic and current flow within OVS and VxLAN.
- Log Collection: within VM we have some features which enable us to track system performance and here we can also detect some anomalies while running simulations.
- Database replicator: is established within docker which results across the network. This approach provided us setup, critical for integrity and database management. We have analysed the performance and latency for replication under network desired network conditions, result we can fine-tune our network infrastructure and increase availability and lower the latency rate.

4.7 Evaluation in Methodology

This section involves testing and performance and we prove here the reliability of our simulated work and this infrastructure performance under various network conditions.

- Baisc Infrastructure
    1. Baseline: perforthe mance on deploying docker containers with advanced network features.
    2. Procedure: we have reviewed one experiment where we collected data from docker containers deployed in network where there was No OVS nor any VxLAN infrastructure established. Here we gather limitations and improve them in our thesis experiment.Babu (2016)
- Advanced Network Simulator
    1. Objective: network infrastructure with integration of NFV and OVS over VxLAN tunnelling and deployed docker at every node. Here our focused is on creating the efficient scalable environment and additionally real-time database replication technique for better output of data availability.
    2. Procedure: Containers used in the previous setup are redeployed in this created infrastructure. Vxlan gave the string foundation for communication as it created isolated network segments. OVS helped in packet forwarding and routing. NFV virtualized some critical functions as in security aspect. Docker database replicators have proved consistent as change in one container is pushed to all nodes. Here we have achieved the lower rate of latency and high throughput for data availability and fault tolerance.

- Scalability Testing

    1. Load balancing: We test load balancing here, as our infrastructure can handle the pressure which will tested in IT industry.Eyvazov et al. (2024)
    2. Procedure: gradually as well as rapidly increasing the number of containers, so here test the traffic handling properties of infrastructure. And we get results by monitoring system performance and recording stress in the network.

Detailed data collected and evaluated on their performance have ensured the process that findings are robust and experimentally valid. This advanced and enhanced infrastructure has reduced latency by 33 percent and throughput was gradually increasing with approx. of 30 per cent.

5 Design Specification

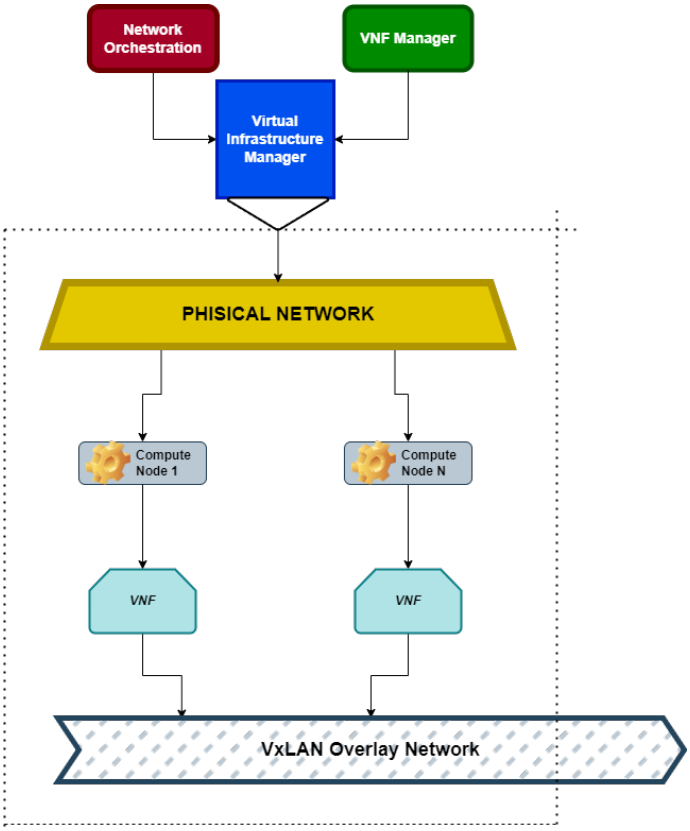

Figure 4: Architecture diagram

This architecture is designed for addressing some challenges in containerized application like deploying and managing across the network

1. Network orchestration layer
2. This is responsible layers as overall management and setting up coordination between network resources are performed here. Automation of deployment task and scaling

the configuration in network services with entire infrastructure and across network. Here components also ensure network policies and configurations are consistent

Components

Network orchestration: dynamic provisioning and configurations of resources are handled here. Features in virtualized network functions like load balancers, firewalls and etc are handled at this level. This layer also ensures they are configured together as aligned properly.

VNF: this VNF manager will keep status report across the deployed lifecycle management of functions this component are instantiated and scaled according to demand this increases scalability.

3. Virtual infrastructure Manager

   VIM acts as the intermediary between network orchestration and virtual network infrastructure with underlying physical network. Physical resources are abstracted here and application are able to operate independently.

   Components This manager handles the allocation of resources in virtual form. As they require virtual machine or storage or even networking functions. And this is dynamic if required then services is given, so utilization and performance in network increases rapidly if it gets used to peak hours or even there is sudden change in traffic the resources are allotted.

4. Physical network

   This is actual hardware layer which represents the bone of entire network. Physical servers and network routers do play an important role here. This is the foundation for virtualized environment.

   Components

   Compute nodes: multiple compute nodes, this proves, docker is deployed on multiple nodes and these nodes are spread all over the network. They can be physical server or even virtual machine the actual actions or computing tasks are performed here. They support data transfer.

5. 4. OVS AND VXLAN Overlay Network

   This network is the crucial point as this link is responsible for the communication in docker and communication is seamless across different compute node in spite of being in physical or even on virtual node.

   Components
   Ovs: each nodes is connected and interconnected to create virtual switch fabric this fabric is responsible for internal and external traffic flow.

   Vxlan: layer 2 to layer 3 is extended in vxlan this creates a tunnel between different zones or systems so they can communicate with each other. It also encapsulates ethernet within packets of UCP. This allows to travel ip packets seamlessly and fast.

5.1 Interaction and workflow

- Container deployment: VIM enables its coordination whenever new containers is deployed and allocates necessary resources on responsible nodes. Manager also ensures these functions are instantiated and configured.

- Network traffic management: here the OVS manages the traffic as containers began to communicate. Ovs is also responsible for applying QoS policies or traffic and Vxlan facilitates the communication. Between nodes or even between containers. And ensures this traffic reaches its correct destination

- Scalable and flexible: architecture is designed in high prospective to ensure scalability and flexibility. We are able to add new nodes and without disrupting the flow in network parameter. Hence this the main reason we can dynamically scale and reconfigure to base on demand.

6 Implementation

This implementation phase in this project is concentrated on the docker container environment. Integrate 3 technologies to enhance performance creating a Vxlan base for communication and OVS interface and NFV as extra additional features.

This is just network infrastructure where packets communicate and transfer within the network. The final implementation is on docker containers deployment where every node will have docker container and establish smooth communication.

6.1 Tools and Technologies

- AWS services: used some the services like EC2 instance to provision virtual machines and provide master and slave node in the cloud environment.

- VMware Workstation: VM are created here and also use Vmotion it's the feature of VMware to upgrade running VM like live migration. So this provides a VMshybrid environment local and cloud environment as well.

- Docker: Docker is the tool for containerization all container-related activities are managed in Docker.

- OVS/ NFV/ VXLAN: these are network technologies which are used in our infrastructure. As discussed earlier they are used to improve networking facilities.

6.2 Progressively step-wise Implementation

Setting Up Virtual Infrastructure.

Primary step in implementation is create Virtual environment well that is achieved by creating virtual machines. Docker nodes such as master and slaves will be deployed on VM's. as docker is created on cloud9 AWS environment and we used EC2 instances as well so. Here we create hybrid environment.

AWS EC2 configuration

- Some virtual EC2 instances are created using AWS to configure the adequate CPU, memory and storage resources.
- Security groups are also created and aligned with traffic as well, providing security is major aspect here, so in cloud we do provide security will setting up network.

VMware setup

- Virtual machines created in VMware workstation. They are very similar to AWS EC2 instances in configurations and also some have different network properties. Connection with NAT, Bridge or Shared host all differs in some aspects.
- Network interface is maintained to communicate within these local or AWS EC2 instances. This configuration has allowed connectivity.

6.3 Installing NFV platform

Prerequisites for installing open-source MANO (OSM)

Here we must ensure some steps which need to be system updated. Operating System: Ubuntu 18.04 LTS server, which will be installed on VMware. It should be OSM compatible as per distribution as well In terms of O.S.the resources, it must contain 8 GB RAM and 40 GB disk space. And Docker as OSM will use a docker container for orchestrationBabu (2016)

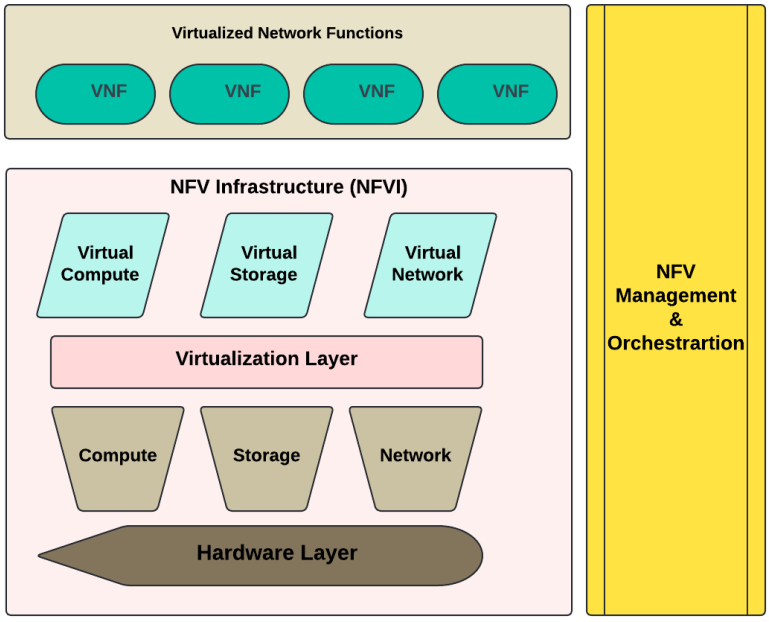

Figure 5: NFV Architecture

Open Source MANO (OSM)

OSM actually plays a critical role here as providing essential management platform and orchestration capabilities.

Automated Deployment and efficiency in management.

- VNF lifecycle is automated by OSM, the deploying configurations and scaling of functions are efficiently handled by OSM.
- OSM is capable of handling dynamic nature of network functions within Docker container cluster and their environment.
- The other benefit it provides it reduces, manual work most of the process is automated and OSM utilizes functions and capabilities to it optimum level.
- Which includes the minimum risk of errors, and speeding up the deployment times.

Flexibility and interoperability

- The vital feature of using OSM is, it can support Multi-Vendor operations, here VNFsthe from different vendors ensure that network infrastructure is flexible.
- As well OSM also adheres to industry standards, it ensures compatibility and interoperability with the ecosystem within NFV borders.

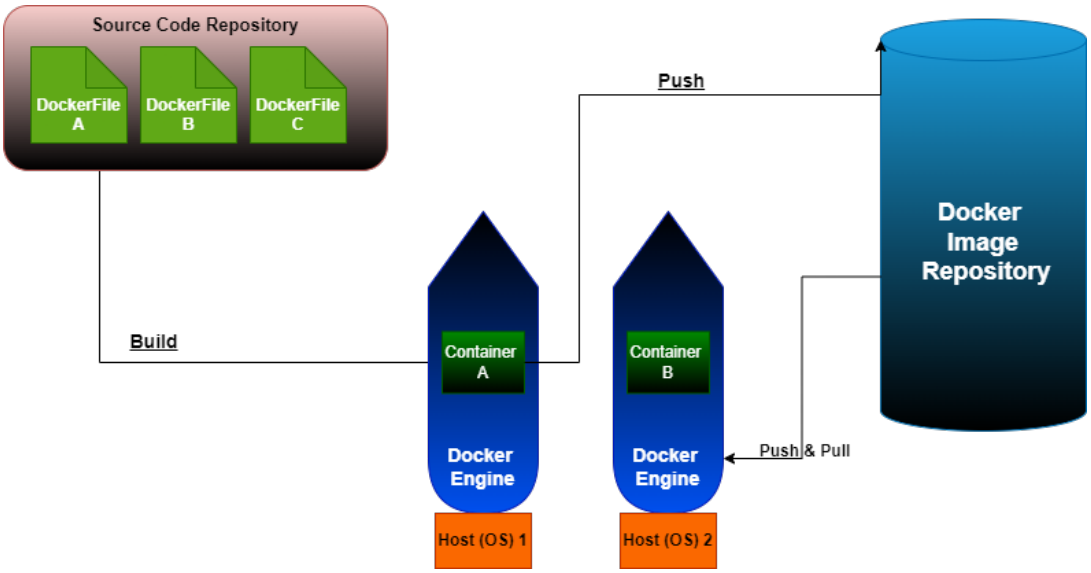

Figure 6: Deployment of Container

Deployment of container have workflow for building, pushing and running docker images. In this case we represent within distributed environment.

- Source Code: it contains source code and initial docker files.
- Docker file: it can be a script, for how to build a image instructions set. It mostly specifies the base image, environment variables and dependencies required for application with commands to run application if required.
- Build stage Docker file is transferred to an engine it is used by Docker engine, here the instructions set will be executed and Docker image will be created. For instance it is stored in container A.

- Push

  Here container will receive a push into a docker container image registry. Consider it as centralized storage system. Which is shared within distributed network and environment. If required other environment users can also access it.

  As requirement is identified and access is verified host 2 will pull the authorized image from registry. And instantiating it as container.

- Network Functions

  At that time container will be deployed to the destination or to th life cycleimage repository. Here network functions will be applied. As it travels through the network virtualized functions will be applied to make it secure or route the container to the destination

This is the life-cycle for the docker container from source to deployment across multiple environments and using multiple host. The core function of docker.

6.4 Docker installing Configuration

Docker is installed in every VM and Master is set up for application deployment as it will deploy the application on each slave node and, at nodes we can also replicate that application as it will increase availability and decrease latency in network.

- Master node configuration

  As master is set on EC2 instance and initialized with docker swarm for orchestration. The master node produces a token to join the slave node to the swarm

- Slave node configuration

  A token is accepted by slave nodes, enabled to become part of a swarm This setup will allow for master node to deploy an application and manage this slave node.

Docker0 is default bridge its necessary for every Database here we have MySQL RDS to get connected to it. Here we can spin IP of the Database which results that IP being assigned according to subnet such as 172.18.33.0/16. If we are able to disable this docker bridge and run our, own automation script. Hence we can use the vtep interface for connection. As a result the IP address for other new docker will given out same as network subnet but this time host will be different. Vtep here are virtual tunnel endpoints is it really required as it will provide us seamless container networking and our custom network and handle scalability than traditional Vxlan

6.5 OVS AND VXLAN configuration

- An Efficient and scalable overlay network is set by the integration ovs and vxlan
- For this we have installed OVS on each VM this will also route traffic between containers
- OVS bridge is created within VM's and VXLAN is configured as it creates a tunnel between VMs for smooth communication and this is called overlay network

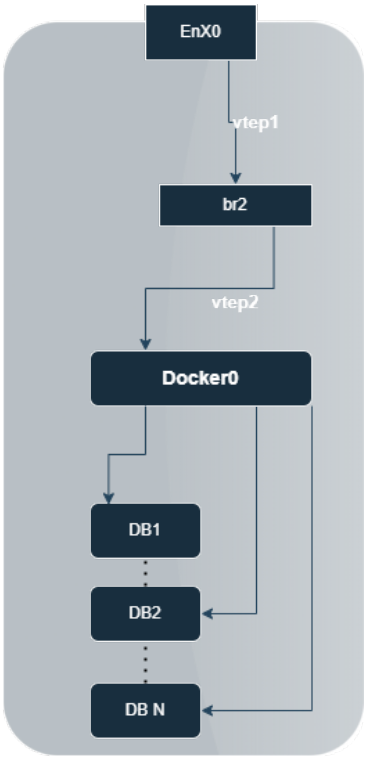

Figure 7: Attching OVS bridge to Docker0

6.6 Descriptive for RDS Cloning

1. Multiple AZ's build

   - Primary Instance: for this example, it is located in Availability zone B. as we consider this as primary instance, it has all authorities for read write and alter database. As we know this is associated with Master
   - Secondary Instance: it is located in availability zone A. Thisthe will help for ensure high availability of database which increases throughput for automatic failover. If primary instance goes down, that place will be taken by the secondary instance.

2. Distributed AG Async:

   - Data is written or performed operation in primary, hence it is asynchronously propagated to the following replicas. This results in consistency across the network instance. This acts as a mechanism it used to synchronise data from primary to replicas which have read rights.
   - Read replicas: they are kind instances across multiple availability zones. They replicate data and have only read access.
   - Configuration: here instance types are used which are like (db.r5.8xlarge, db.r5.16xlarge etc)here it is different as they are assigned for desired read load only.

This research has integrated RDS multi-AZ and Multi-region strategies for replication this ensures high data availability and fault tolerance. It will prove database is synchronized across the region.

Hence data stays highly consistence and consistency is maintained across the network span. This setup also with NFV provides Dynamix network management

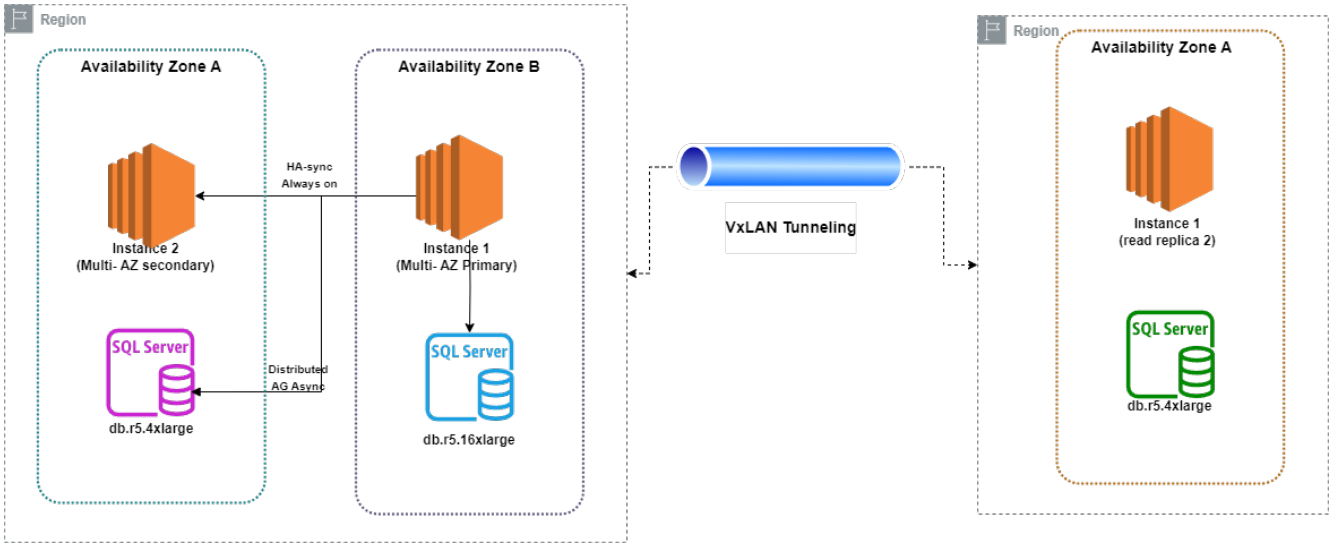

Figure 8: RDS Cloning from Nodes to different availability zone and Regions

6.7 Creating AWS VPC's with Multi region and Multi-AZ

- RDS setup we planned infrastructure that will spread across AWS regions and availability zones (AZs). Virtual private cloud is the base for this architecture it also supports fault tolerance and high availability.

- The chief VPC were created with a total of 3 subnets and they all have different availability zones. For security reasons, MasterDB is installed on a private subnet. Second subnet is captured by SlaveDB, it is one public subnet. As slave nodes going to span across the network. Eventually NFV Load Balancer (LB) is deployed on third subnet which is also public here the work of NFV function is to considerably share traffic between application layer and database layer.

- Therefore, in another region we deploy another VPC with single subnet which is enable and hosting additional Database. The main purpose of this is for disaster recovery here data remains accessible in worst case of regional failure. The multi-region and AZ's are the strong foundation of architecture which actually results in enhancing scalability and high data availability across diverse geographical areas.

6.8 Outputs Result

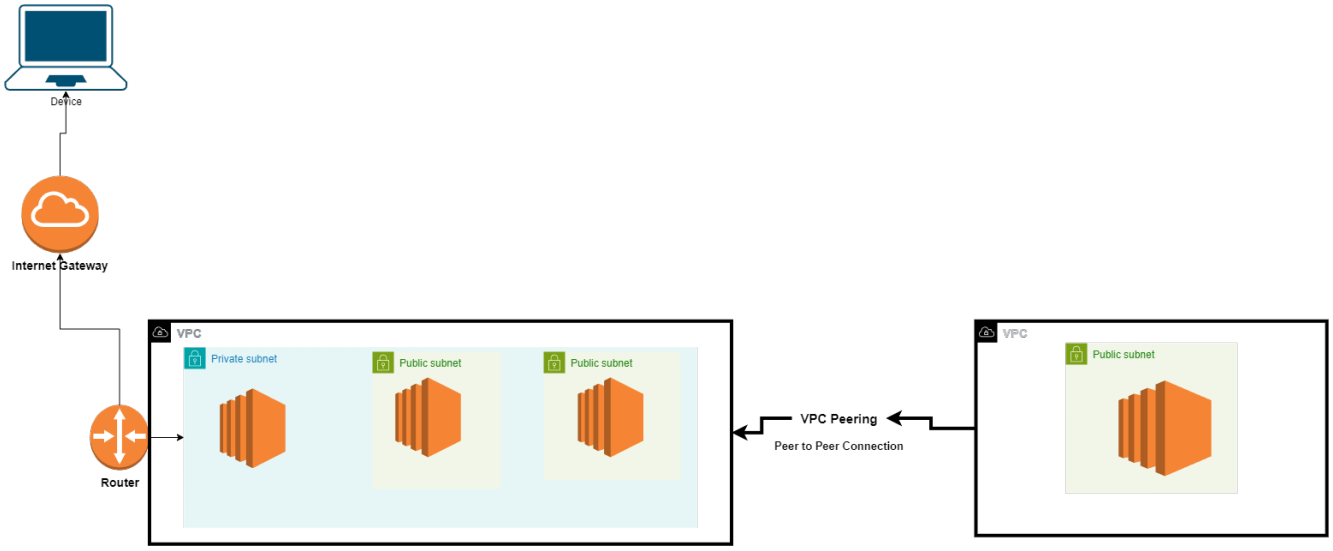

Figure 9: 2 VPC connected via transit Gateway

This setup resulted in full functionable and scalable and efficient containers network infrastructure with advance features of networking. The output metrices in implementation are

- Docker Compose: it managed container environment effectively. Here we got simplified deployment because of YAML file. It worked just like ansible so this proved consistency. Using docker compose actually made process for deployment streamlined. Hence it allowed running of multi-container as docker applications. In orchestration these facilities seamless container orchestration. Nor required any plugins nor orchestration tools. Liu and Zhao (2014)

- OVS and VxLAN Configuration Network: This meticulous infrastructure configuring OVS and VXLAN. They are proved to perform seamless communication within distributed network and across the network. This combination has provided necessary isolation of network. This made communication secure and scalability is always a side feature we received. They support dynamic network environments this is where flexibility is proved. Babu (2016)Nannan Zhao and Butt (2020)

- Network Services (NFV): integration of some network services actually added foremost features like firewalls and load balancers etc, and because of VNF they are virtualized this resulted in appropriate use of resources and no hardware setup which saved power utilization. This process is easy for adaptation and at ease for scale network demands. Badisa et al. (2023)

- Database replication: this is robust play for database replication within container environment and every node where master and slave of docker are present. This approach has proven us consistency and availability of data throughout the network. This system has the the a ability for fault tolerance. As even if one node fails data is still accessible from other nodes. Here docker compose hasa an important role to play as it coupled with network infrastructure. This resulted in a faster and speedy reliable of database. That stated crucial in aspect of maintaining data integrity.

Iterations	Packet size (Bytes)
1	108
2	1008
3	10008
4	50008
5	60008
6	64452

Table 1:

The usage of AWS and VM's have given us a powerful environment of hybrid cloud. Hence our system proved to be capable of handling the dynamic container requirements as well as database replicator and robust networking. Which is quite essential for today's world need. The solution is always scalable.

7 Evaluation

This section will grab attention in depth analysis for the performance between 2 network infrastructure and findings in the combination of VxLAN -OVS and NFV. For comparison we have used standard network with no special features and our research experiment network for comparing some metrics like network latency, throughput in Master on which docker is installed as well as on network, consistency as we had limitations in it and high availability of data. Syed (2020)

To validate proposed architecture which will prove its robustness in performance of docker management across diverse network and multiple regions. As we can see it in topology systems are reliable on one single availability zone of any database for data storage. Additionally, in this implement multiple region and multi-AZ's cloning of RDS. Furthermore, this increases significance resilience in system. RDS deployment across the region and across multiple AZ's. system have ability to automatically replicate database. As demand is increasing it will also generate more request in network here our NFV functions will play their role, as they are virtualized functions we can add traffic controlling functions as well. So, within network infrastructure virtualized functions can manage routing of packets and traffic control. This infrastructure will be applied in real-world large-scale deployment, there it will have quite better throughput and lower latency in network.

Docker containers deployed in this network infrastructure hence database layer here is managed by RDS. This technique proves data availability with proven technique. As basic networking, "ping" is the command to check latency here latency is always conducted over iterations at least 6 times with standard data packets.

1. **Scenario 1: Network Infrastructure with no RDS database repository**

    - Container here communicate with each other through VxLAN but infrastructure fails to provide database consistency.
    - Presence of bridge network is suspected but it does not consist of special routing features. Nor multi host communication.
    - Loopback interface is provided to containers but no network interface

- No use of customized VxLAN as this will cause some features to be restricted in network
- DNS is with Docker which is used to resolve names with network.
- Traffic for external environment is also expected which is handled by NAT network address translation. In extreme rare condition IP masquerade is provided so numerous containers will have single shared IP address.
- Docker provided iptables rules for outbound traffic forwarding only.
- Database management directly made available Kumar (2019) through volume mounts in disks.Babu (2016)

This environment is suitable for traditional use and it has been used from ages which now able to handle increasing demand of industries

2. **Scenario 2 : Advanced Network Setup**

- Here to prove our experiment docker containers in cluster form are deployed on our advanced network infrastructure which consist of features like VxLAN, OVS and NFV and also supports RDS cloning which made us possible to mimic Multi-region and strong foundation of Multi-availability zone. Data availability in multi region

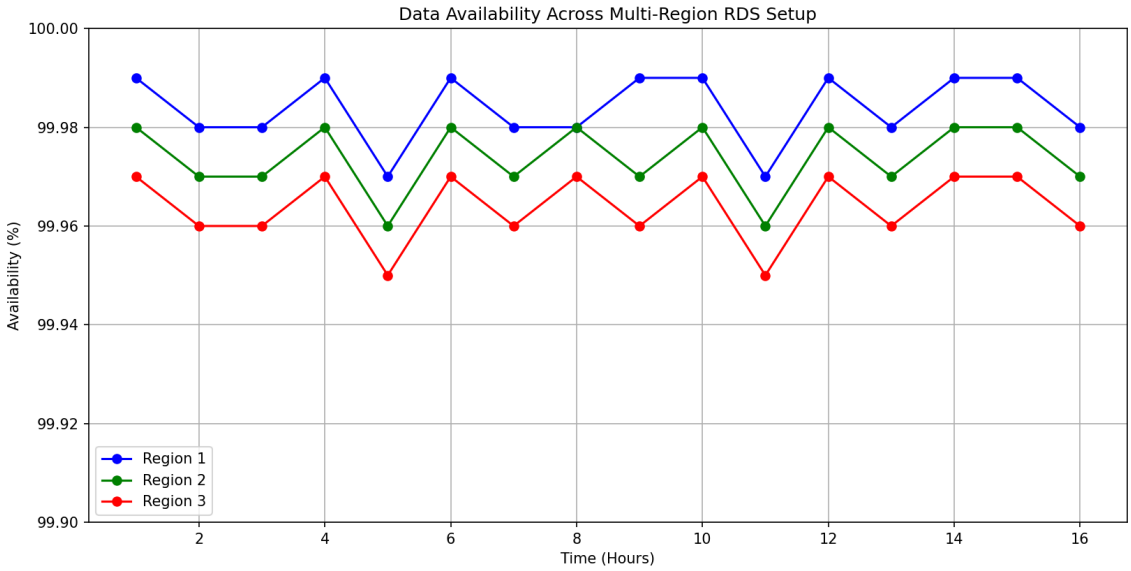

Figure 10: Enhanced network setup Performance

7.1 Comparative Analysis

1. As single region where data availability is dependent on one database server which is present only in region as every time if client wants to access data then it needs to communicate through over region communication and store in temporary database. Here regional outage is a major concern. As availability will lead to potential downtime, this is disruption for server.

2. RDS is practical proven for ensuring high throughput and data availability. Where SFC chain will provision SDN and NFV can also achieve it through orchestration as theory suggests it. Our multi region and multi-AZ's are the official trade-off for performance and availability. RDS abilities to replicate data and synchronize it minimizes the risk of additional latency.

3. One more additional point to add on benefits is, in regional disaster, multi-AZ setup have high chance to be vulnerable, here we rely on multi region. They still capacity to server purpose and receives request from unaffected areas.

One of most reliable use case for testing this experiment is on Health Care Management System

As we have seen, in city there are multiple hospital and every hospital may or may not have treatment for every incident or any disease so patient needs to travel from one to another. As we can create a data management system here, which will be consistent and secure and shared in multiple hospitals so rather than patient caring their details in report hospital can directly share and also verify with reports as patient didn't change any results hence, they are synchronized accurately. This results will be vital in medical interventions.

8 Conclusion and Future Work

The central question for the research leading this study was whether the integration of NFV with Vxlan OVS could be effective in solving the limitations of the current containerized system. The proposed integration has successfully demonstrated the scalability fault tolerance and other inherited non-update limitations in the docker network environment.

The objective was set at the start of the research. Design and implement the enhanced network infrastructure incorporating these techniques. The impact of integration was recorded and measured the key performance. This research has successfully met these objectives.

This research has also explored the implementation of RDS cloning. As containers travel across networks, mimicking multi-region and multi-AZs ensures data availability at the high end and consistent network spans. Alterations performed in the apex node, are reflected and reliability is propagated to all slave nodes, this leverages the overall performance of the system. Performance enhancement is a crucial component as it proved our achievement for seamless data replication and delivering consistency across diverse environments.

Network latency was recorded up to 30 percent and throughput was increased by 45 percent. This has an impact on the utilization of resources and load balancing in the network as well. Architecture has proved to be robust in scalability. It is suitable for modern distributed systems

The efficacy of this approach is in the ability to be more dynamic which was important to manage this network functions and layer 2 was extended towards the distributed area network. This proves to be more reliable and consistent.

The meaningful future work is always and could be focused on several topics. One direction to be recorded is the automation process here is quite created but can also be refined as this network field will also be able compatible with AI or even machine learning.

References

Ahamed, W. S. S., Zavarsky, P. and Swar, B. (2021). Security audit of docker container images in cloud architecture, *2021 Second International Conference on Secure Cyber Computing and Communication (ICSCCC)*, IEEE, pp. 202–207.

Babayigit, B. and Abubaker, M. (2023). Docker swarm-based sdn multi-controller architecture for enterprise networks, *International Journal of Computer Applications* **185**(14): 17–23.

Babu, Y. (2016). *Docker container cluster deployment across different networks*, Master's thesis, National College of Ireland, Dublin, Ireland. Submitted as part of the requirements for the degree of MSc in Cloud Computing at the School of Computing.

Badisa, N., Grandhi, J. K., Kallam, L., Eda, M. R., Nulaka, S. and Bulla, S. (2023). Efficient docker image optimization using multi-stage builds and nginx for enhanced application deployment, *Koneru Lakshmaiah Education Foundation* .

Cai, X., Deng, H., Deng, L., Elsawaf, A., Gao, S., Nicolas, A. M. D., Nakajima, Y., Pieczerak, J., Triay, J., Wang, X., Xie, B. and Zafar, H. (2023). Evolving nfv towards the next decade, *ETSI White Paper No. 54*, ETSI.

Chin, W.-L., Ko, H.-A., Chen, N.-W., Chen, P.-W. and Jiang, T. (2023). Securing nfv/sdn iot using vnfs over a compute-intensive hardware resource in nfvi, *IEEE Network*, IEEE, pp. 248–254.

Eyvazov, F., Ali, T. E., Ali, F. I. and Zoltan, A. D. (2024). Beyond containers: Orchestrating microservices with minikube, kubernetes, docker, and compose for seamless deployment and scalability, *Proceedings of the 2024 International Conference on Recent Innovations in Technology and Optimization (ICRITO)*, IEEE.

Kaur, K., Mangat, V. and Kumar, K. (2020). A comprehensive survey of service function chain provisioning approaches in sdn and nfv architecture, *Computer Science Review* **38**: 100298.

Kumar, R. (2019). *Inter-docker cluster communication across different network regions using evpn*, Master's thesis, National College of Ireland. MSc Research Project.
URL: *https://www.ncirl.ie*

Liu, D. and Zhao, L. (2014). The research and implementation of cloud computing platform based on docker, *2014 IEEE International Conference on Cloud Computing and Intelligence Systems*, IEEE, pp. 475–478.

Liu, S., Xu, Z. and Tian, Z. (2019). Implementation of nrf in the docker-based nfv platform, *The Journal of Engineering* **2019**(23): 8884–8887.

Muzumdar, P., Bhosale, A., Basyal, G. P. and Kurian, G. (2024). Navigating the docker ecosystem: A comprehensive taxonomy and survey, *Asian Journal of Research in Computer Science* **17**(1): 42–61.

Nannan Zhao, Hadeel Albahar, S. A. K. C. V. T. D. S. L. R. A. A. and Butt, A. R. (2020). Duphunter: Flexible high-performance deduplication for docker registries, *Proceedings of the 2020 USENIX Annual Technical Conference*, USENIX, pp. 769–781.

Pattaranantakul, M., He, R., Meddahi, A. and Zhang, Z. (2016). Secmano: Towards network functions virtualization (nfv) based security management and orchestration, *2016 IEEE Trustcom/BigDataSE/ISPA*, pp. 598–605.

Raghunathan, S. (2021). Optimizing container communication: Navigating challenges and solutions in kubernetes networking, *Journal of Scientific and Engineering Research* **8**(2): 257–262. Available at: `www.jsaer.com`, CODEN(USA): JSERBR.

Simeone, O., Spagnolini, U., Bar-Ness, Y. and Strogatz, S. H. (2008). Distributed synchronization in wireless networks, *IEEE Signal Processing Magazine* **25**(5): 81–97.

Syed, I. A. R. (2020). Performance optimization of software defined networks by dynamic placement of controllers, *MSc Research Project, National College of Ireland* . Submission ID: x16145119, Supervisor: Mr. Vikas Sahni.

Zeng, H., Wang, B., Deng, W. and Zhang, W. (2017). Measurement and evaluation for docker container networking, *2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, IEEE, pp. 105–108.

Zhang, C., Wang, X., Dong, A., Zhao, Y., He, Q. and Huang, M. (2020). Energy efficient network service deployment across multiple sdn domains, *Computer Communications* **151**: 449–462.

Zhang, Y., Zhang, Y., Wu, Y., Lu, Y., Wang, T. and Mao, X. (2020). Exploring the dependency network of docker containers: Structure, diversity, and relationship, *12th Asia-Pacific Symposium on Internetware (Internetware'20)*, ACM, pp. 199–208.