Expertise
and insight
for the future

Mutale Chewe

Hybrid Cloud Infrastructure Security

Security Automation Approaches for Hybrid IT

Metropolia University of Applied Sciences

Master of Engineering

Information Technology

Master's Thesis

8th June 2021

PREFACE

Full-time work and graduate studies is never an easy undertaking, but this had to be done. I was driven to succeed as a way to payback everyone who ever believed in my abilities. I one day hope my academic pursuits will motivate my son Ryeon R Chewe to do even greater things than I have done before he's 25.

I thank all the people who made this thesis possible, for their collective contribution.

Helsinki, May 25, 2020
Mutale Chewe

Author Title Number of Pages Date	Mutale Chewe Hybrid Cloud Infrastructure Security 52 pages + 3 appendices 5 June 2021
Degree	Master of Engineering
Degree Programme	Information Technology
Instructor	Ville Jääkäläinen, Principal Lecturer

Security is the single most consequential public cloud adoption barrier for enterprise. The need to extend on-premises IT infrastructure to public clouds poses great security challenges. Solution architecting to the said challenges is critical both technically and business risk wise. Implementing infrastructure security in hybrid IT with a shared responsibility security model requires architecting with ever evolving technologies. The challenge lays in defining security principles to guide the security architecture, which can be operated across two or more clouds with different security approaches.

This thesis researched security approaches to architecting hybrid cloud security, by evaluating security implementations and coming up with recommendations on security posturing. An analytical approach to architecting hybrid security was used. The resulting security recommendations can be as a reference guide when implementing and managing hybrid infrastructure security. In the final analysis, the role of identities and access management is advanced as a step towards orchestrating and managing security by code within the context of infrastructure as code for securing hybrid cloud infrastructure. The methodology includes describing a security approach for orchestrating, automating and managing hybrid security. The study hypothesis is that security technologies are only as good as the architectural principles and approach upon which they are built and applied.

Keywords	Hybrid IT, Cloud, Security, Automation, Architecting

Contents

List of Abbreviations

AI	Artificial Intelligence
AWS	Amazon Web Service
CAPEX	Capital Expenditure
CSA	Cloud Security Alliance
DDoS	Distributed Denial of Service
IAM	Identity and Access Management
Infra	Infrastructure
IP	Internet-Protocol
IaaS	Infrastructure as a Service
ISP	Internet Service provider
I.T	Information Technology
IoT	Internet of Things
IAM	Identity and Access Management
LAN	Local Area Network
NCC-SRA	NIST Cloud Computing Security Reference Architecture
OSI	Capital Expenditure
OS	Operating System
PaaS	Platform as a service
RBAC	Role Based Access Control
RDP	Remote Desktop Protocol
SaaS	Software as a service
SSH	Secure Shell Host
SIEM	Security Information and Event Management
SASE	Secure Access Service Edge
SOAR	Security, Orchestration, Automation and Response
VM	Virtual Machine
VPC	Virtual Private Cloud
VPN	Virtual Private Network
YAML	YAML Ain’t Markup Language
WAN	Wide Area Network

1 Introduction

Hybrid cloud security is defined by Red Hat as the protection of data, applications, and infrastructure associated with an Information Technology (IT) infrastructure that incorporates some degree of workload portability, orchestration, and management across multiple IT environments. ISO/IEC 17788-20144 defines a hybrid cloud as a cloud deployment model that uses at least two different cloud deployment models. Gartner one of the leading IT research and advisory firms in its 2017 technology trends report, predicted that 90 percent of organizations would adopt hybrid infrastructure as a computing model by the year 2020. Private IT also known as on-premises, requires a secure extension strategy for enterprise infrastructure. Hybrid IT is increasingly becoming a default computing model for modernizing legacy infrastructure. The aforementioned is partly driven by new computing architectures for deploying applications. These computing infrastructure architectures includes containers and big compute among many cloud hosted infrastructure or platform services. As illustrated in Figure 1, the integration of public cloud infrastructure with traditional IT (on-premises/private IT) is what is referred to as hybrid IT or cloud. Despite all the benefits of a hybrid cloud deployment model, security is the foremost challenge which requires architectural solutions to support implementation, orchestration and management.

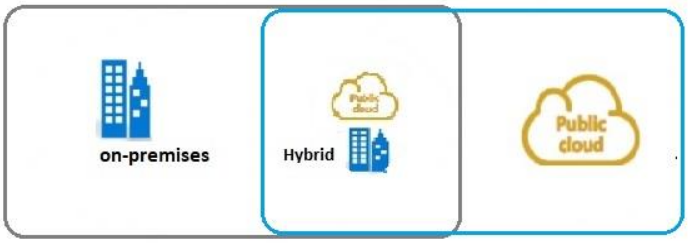

Figure 1: Schematic illustration of a hybrid cloud

A hybrid cloud infrastructure as shown in Figure 1, couples the private (on-premises) infrastructure with public cloud hosted infrastructure. The immediate challenge to address become how to securely integrate infrastructure hosted in public cloud with private infrastructure. Security in private IT is designed and operated by the enterprise. In public clouds, the security responsibility is shared with the provider who owns the computing servers in commercial data centres operated as multi-tenant hosting infrastructure.

1.1 Background

The growth and shift in enterprise IT towards hybrid and multi-cloud deployment model as predicted in the 2019 Gartner research on cloud adoption trends has been realized. Integrating Private IT compute infrastructure with public cloud virtual infrastructure in the absence of effective hybrid security solutions presents data and application exposure risk. It can be surmised that architecting infrastructure security to secure networks, critical business applications and data is critical for a successful adoption of hybrid IT as an Infrastructure strategy. The research into an effective approach of architecting hybrid cloud security was proposed by a large telecommunication company, with an aim to evaluate and develop a security approach of securely extending on-premises compute infrastructure to the public cloud. The aim was to evaluate hybrid cloud security strategies and solutions advanced by leading public cloud providers and advising the best approach. Gartner's 2019 research report on cloud trends placed Microsoft, Amazon and Google as leaders. The subject enterprise managed a large mixed private IT deployment with a VMware private cloud hosted on-premises alongside an extensive legacy infrastructure built over many years. Table 1 below outlines the requested research deliverables.

Table 1. Enterprise Security research goals and sort deliverables.

Security Model	Cloud Connectivity	Security tooling
Which security model would best meet security requirements for building Hybrid IT Infrastructure? Deliverable: Documented infrastructure security design guidelines' and recommendations describing the security approach, policies and best-practices for operating hybrid IT(Cloud)	What technologies best secures our private infrastructure integration with the public cloud. Deliverable Description of technologies and methods for seamless integration of public or hybrid infrastructure external to an enterprise infrastructure	What tools do we need to securely deploy infrastructure and manage our hybrid infrastructure at scale. Deliverable Outlining Infrastructure security orchestration and management tools recommendations.

Important for securing the private to public IT integration, is the Layer 3 interworking of private cloud and on-premises infrastructure being the principle security concern. The question being how best to interlink the perimeter gateways on either sides of data centers, whether by VPN or dedicated private connection between the virtual networks in the public cloud and the local data center networks and resources.

1.2 Rationale and Research scope

The rationale can be summarized as developing a production ready hybrid cloud infrastructure security orchestration and management approach. Central to the research, was evaluating how well documented provider security reference architectures and best practice recommendations meet enterprise requirements for production ready deployments. Figure 2 illustrates the research scope using a defence in depth representation model of cyber security. Physical security is a provider's domain in public clouds.

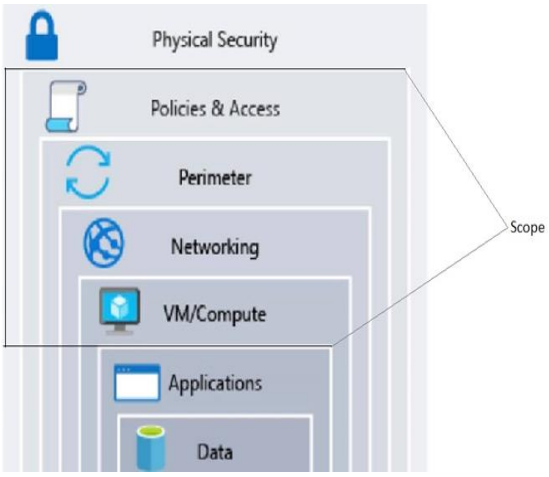

Figure 2: Security scope illustration using layered security approach [Microsoft Azure]

The layered approach illustrated in Figure 2 is a defence in depth security pattern in IT security which is recommended for both traditional and cloud based Infrastructure as a Service (IaaS), whose elements includes; networking, storage, compute and support services such as identity and access management (IAM).

1.3 Research Question and Approach

The research question was formulated as follows: Do cloud provider reference architectures and best practice recommendations meet enterprise security requirements for production ready hybrid infrastructure deployments? The study aims to answer the stated research question in respect to goals listed in Table 1 determining the best approach for architecting hybrid infrastructure security at scale, comparable to on-premises security levels as a reference standard for baseline security requirements.

The study takes a comparative approached in implementing hybrid security by evaluating architectural integrity and technologies against proven security implementations. The above stated approach is evaluated against cyber security principles and hybrid security recommendations. The hypothesis is that hybrid security implementations are only as effective as the architectural principles and approach upon which they are based or how they are applied in a hybrid IT deployment. A consolidated or unified hybrid security approach is an advised approach for implementing a transparent security management across a hybrid environment to enable visibility [1]. A summary list of core security areas covered in this study are here defined as a working list of security benchmarks to be satisfied in the architecture.

- Connectivity and edge security
- Identity consolidation.
- Secure infrastructure access.
- Secure infrastructure configuration.

In practice, developing a hybrid security strategy requires putting together technologies and best practices to fit in a security design (architecture) for securing infrastructure. Understanding hybrid security threats is the first step to building a resilient hybrid security solution. The configuration and security solutions are premised on defined security baseline and security requirements. The security baseline is often based on an organization's IT security policies, practices, trusted and tested or recommended approaches. These sometimes are driven by an enterprise's security posture and strategy.

1.4 Thesis Outline

The thesis is organized in 7 chapters, starting with an introduction which covers the research background, scope, aim, rationale, and research methodology. The research goal and hypothesis are also stated and explained here. Chapter 2 describes the fundamental principles of IT security and the general approach to infrastructure security. Chapter 3 gives a descriptive current state analysis of hybrid cloud security and its related challenges. In Chapter 4, the thesis discusses the approach to architecting hybrid cloud security solutions. Chapter 5, describes the important topic of cloud security orchestration and automation, which is then followed by Chapter 6 which highlights methods of implementing security management. The thesis is finalized with chapter 7 providing discussion and conclusions where suggestions, proposals and recommendations on how to approach the implementation of hybrid IT security solutions with current tools and technologies while building a foundation for the foreseeable future are presented.

2 General Approach to Infrastructure Security in IT

This chapter explains the general approach to infrastructure security. It gives an overview of both traditional IT security and cloud security. The chapter provides a security overview and highlights the differences in approaches between traditional IT and cloud security. Understanding the different approaches helps in finding trade-offs when unifying the security management for hybrid infrastructure. Table 2 shows how traditional IT differs in comparison with cloud-based infrastructure in its approach to security. What can be deduced from this comparison provides an insight into what integration challenges need to be overcome when implementing or orchestration hybrid security.

Table 2: Contrast in Security Approach between Cloud Native and Traditional Security

Native Cloud Security	**Traditional Enterprise Security**
Automation: Automated response to threats coupled with AI, also supports the adoption of immutable infrastructure which helps eliminate misconfigurations.	**Monitored and Instrumented**. Active monitoring with manual response for threat mitigation.
Proactive: Operate with openness to quick change and response to eliminating threats.	**Reactive**. Threat mitigation occurs after detection and availability of personnel to mitigate the threats after occurrence.
Patched via clean-slate redeployment. Auto patching can be enabled to apply new patches as soon as they become available.	**Patched incrementally**. Patches are applied incrementally by internal security teams often have to be trigged.
Promoting change: Postured to support faster change, aided by automated or CI-CD tools/DevOps	**Resisting change**: Organization is slower to change method.

It can be seen in Table 2 above, that operating cloud infrastructure is an IT transforming exercise that gives rise to a new form of security approach and posture. Whereas traditional IT security takes a passive or reactive approach, cloud and hybrid security is built around an active and responsive security approach that leverages code and automation for effective security management. It can also be seen in Table 2 that hybrid security orchestration adopts new tools and processes like DevOps.

2.1 Cloud Native Security Architecture

Cloud Native Computing Foundation (CNCF) defines “cloud native” as being technologies that, “empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. The said technologies include infrastructure platforms such as containers or immutable infrastructure and declarative APIs. Figure 3 provides a picture of how Infrastructure deployment looks in a cloud setup. If the back-up site has to serve as a disaster recovery set-up, both the infrastructure and security implementation needs to be replicated.

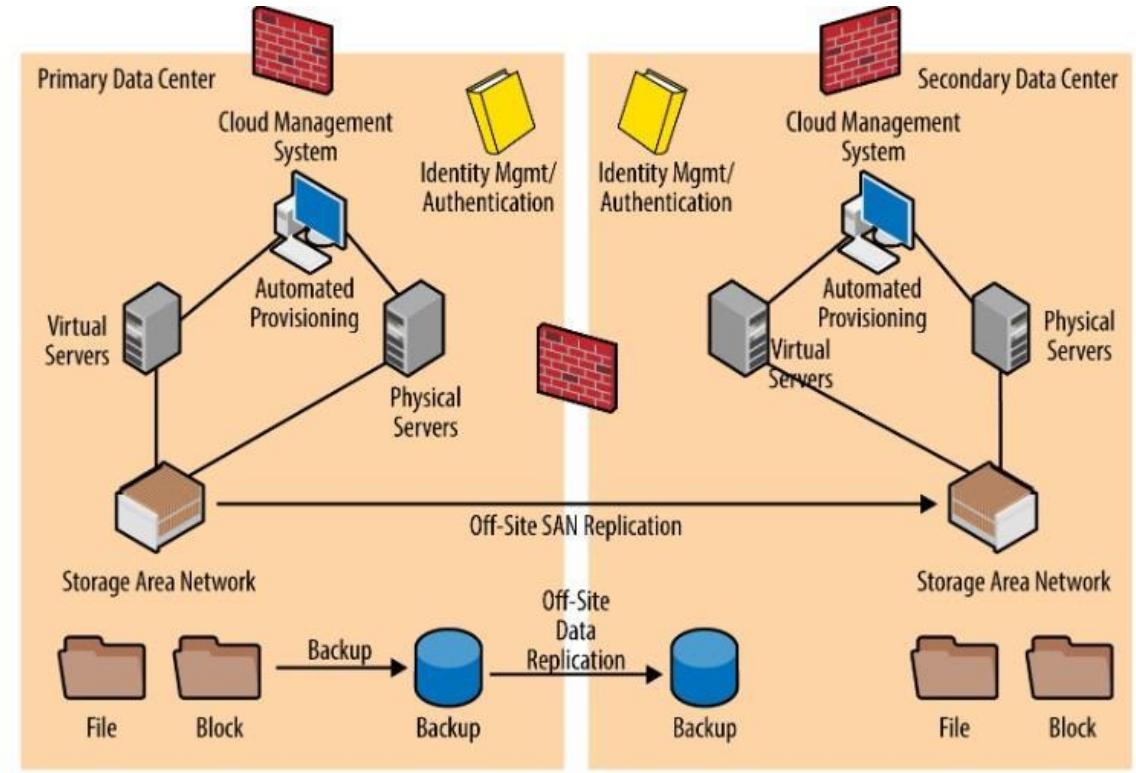

Figure 3: Common IaaS architecture of a cloud deployment [1, 44]

Cross domain security configuration is advised for geo-redundant replication. The different geographies should be segmented into virtual domains to enhance security. Separate certificate authorities for each management domain is ideal and best implemented to ensure secure and private communications between gateways and their management domains. Having illustrated what a common infrastructure deployment looks like, it is possible to define a list of security requirement on a high level, to help prepare a baseline for implementing security from the many different approaches available.

Listed below are the initial requirements for architecting a general template for a hybrid infrastructure deployment.

1. **Management and Automation**
    - Data centric security policies
    - Continues security delivery and controls
    - Operational visibility across the hybrid
2. **Secure Connection and Access Controls**
    - Secure private to cloud connectivity
    - Trust and network segmentation
    - Identity, authentication and authorization
    - Fine grained access control
3. **Immutable Infrastructure and Continuous Integration/Delivery**
    - Cloud native security integration (CI/CD)
    - Infrastructure orchestration by code (IaC)
    - Security automation (management)

The list includes the requirements in summary for architectural guidance. This helps with a high-level definition of security aims. The requirements listed here captures high level hybrid security requirements from which a security approach can be formulated. Organisations need to approach the building of hybrid security from a position where provider security maybe insufficient. This approach is necessary because public cloud security is subject to trusting the security implementation of a provider's multi-tenancy data centre, where verifying the security integrity in the provider's responsibility domain is not possible. Red hat suggests a defence-in-depth strategy, which calls for an integrated and layered security strategy that covers process and technologies [2]. The idea is not to rely on any one security strategy hoping it will always work. After the security strategy has been formulated, it has to be determined what security architecture suits the work-loads to be hosted on the subject infrastructure. This is done by defining a high-level infrastructure component architectural plan. The use case scenarios and architectural patterns can be selected at this point for the purposes of iterating through a proposed topology.

A provider reference architecture needs to be examined to gather the topological construction from a security perspective. The reference design insights can then be used to draw up a suitable overview of the deployment schematic [3]. The architecture from

the onset needs to be conceived around satisfying security benchmarks derived from proven best practices that promote security characteristics such as:

- Consistent security approach
- Process automation (security)
- Single pane monitoring and management
- Common governance and compliance (policies)

It can be seen by contrasting and comparing the characteristics of traditional IT and cloud operational approaches shown in Table 3, why the characterized security approach above, best represents a part of the desirable attributes of a well-designed hybrid infrastructure security implementation.

Table 3: Comparison of Traditional IT Security Principles against Cloud Based

Traditional IT	**Cloud**
Simplified management by aiming to optimize performance, resilience, and availability	Create flexible deployment option across both private and public cloud.
Standardized architecture to optimize performance, resilience and availability.	Maximized scale and IT efficiency for managing workloads across hybrid IT by extending the same infrastructure, operations, tools, and processes across deployments.
Intrinsic enterprise-level security with consolidated solutions for both traditional and modern workload types.	Active and responsive security automated monitoring and security incident response.

It can be deduced from Table 3 that cloud security takes a proactive posture as opposed to traditional IT security which is reactive and slow to introducing system and security changes, which may include security responses or remediation [5]. Implementing hybrid security is best implemented by balancing both public cloud and private IT best practices.

2.1.1 Security Models in Hybrid-IT

After going through the process of identifying and selecting a security approach, the next step is deciding what hybrid security models would best deliver on the requirements. The security architecture has to be built around a security model. There are many general or deployment specific security models. Zero Trust (ZT) security model for example, is an IT security model based on strict identity verification for every user and device trying to access resources on a private network. Encryption is the primary method for securing data at rest or in transit in a Zero trust [4]. With a zero trust approach, regardless whether a user attempting to gain access to infrastructure is within or outside of the network perimeter, they have to be subjected to stringent identity scrutiny. Zero Trust is not a set of tools or technologies but an architectural principle [13]. There are currently three (3) fundamental principles that define a Zero Trust security philosophy (listed below) which enhances hybrid security.

- Never trust
- Always verify
- Always enforce least privilege.

The underlying security idea in zero trust is that infrastructure access must be strongly authenticated and only authorized based on qualified identities for a granular least privileged scope, in which access should be granted to complete only defined operations on an infrastructure resource. This security approach fits into an identity driven security approach that advances identity as the new security perimeter [4]. Unlike in traditional IT networks, zero trust is not based on the “castle-and-moat” perimeter concept, where preventing access from outside the network is the security focus. The latter approach lets anyone inside the network or those who find themselves inside to transverse the network on default trust. The flexibility yet stringent evaluation of identities makes zero trust ideal for hybrid security. Flexibility is required to seamlessly allow an identity to be used both on-premises and in the cloud. A combination of both identities, their security attributes and policies is best implemented to extend conditional access security policies as part of a zero trust posture [5]. The security philosophy behind zero trust is based on a proactive posture that regards every user as a potential security threat. Identities within or outside the network are evaluated in the same way and not accorded unlimited trust in the course of a session.

2.1.2 Approach to Container Security

Containerization is a cloud native infrastructure platform for building immutable infrastructure. Containers have increasingly become an enterprise application deployment means of delivering distributed applications. The benefits of operating immutable infrastructure include the seamless reprovisioning of failed instances of an infrastructure deployment. Modern application architectures like micro-services and telecommunication edge computing are best deployed on containers which remove the need to manage application dependencies at infrastructure level.

Building cloud native infrastructure with container orchestration technologies such as Kubernetes, Open Shift and Docker, requires a process embedded security approach. Important elements of orchestrating container security include securing images, registries and the control plane. Container images in particular hold executable packages that includes everything needed to run an application such system tools and system libraries. Images and registries need being validated by scanning for security flaws, public packages in specific have been known to sometimes have exploitable security flaws. Figure 4 shows a DevSecOps continuous process approach to security.

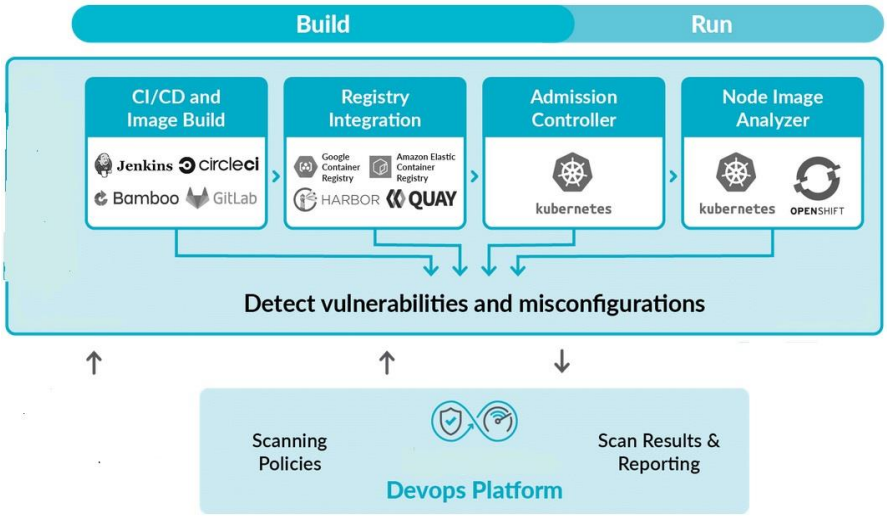

Figure 4: Example Container security orchestration setup

DevSecOps is secure DevOps, a way of embedding security in the orchestration process as opposed to being a day 2 activity. Figure 4 illustrates an approach for setting up container security. It is very important when dealing with container infrastructure, that images be hosted in private authenticated registries in production environments for

obvious reasons of security. For access security, least privilege assignments must be enforced in admission control by leveraging Role Bases Access Control (RBAC) and privileged access management session policies. Tools for validating configurations are required, an example being the Kubectl client can be used for running pre-scripted test suits to verifying Role Based Access Control (RBAC) settings for Kubernetes deployment [10, 172]. Container pods and their access credentials need to be protected and secured. It is advised to secure administrative consoles with passwords and Multi Factor Authentication (MFA) if possible. This will prevent known security exploits. The approach described in Figure 4 on the preceding page can be implemented in a number of steps as shown in Figure 5 below, as a generic example. Container image security is the first security check point when setting up container-based infrastructure. Images are known to be one source of security vulnerabilities.

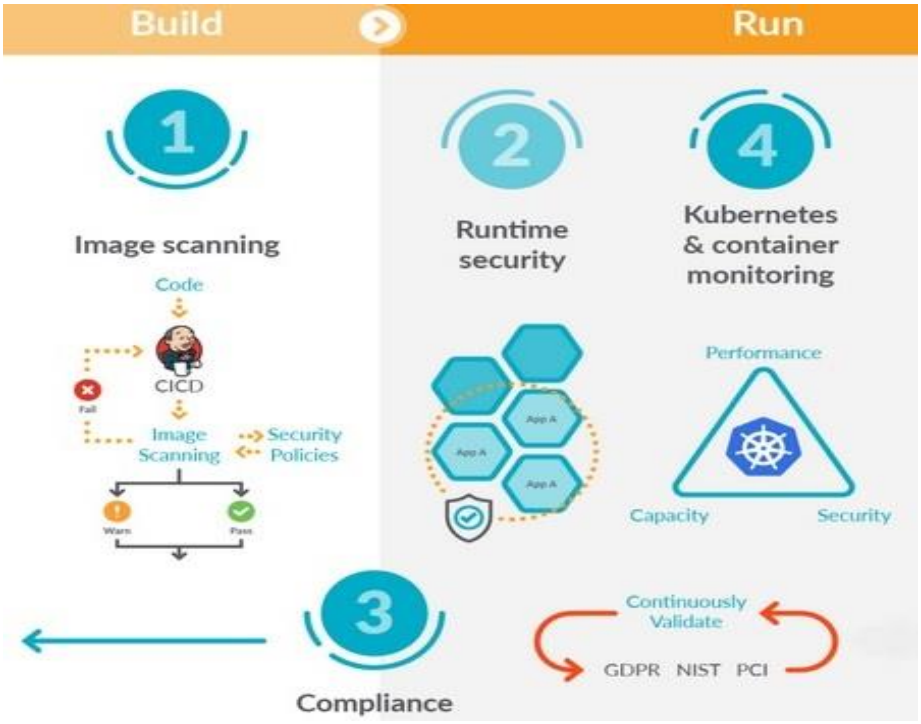

Figure 5: Five (5) step process of implementing of container security.

Restricting access to image repositories is required for enforcing security and ensuring that the images running in a given environment are verified to be security compliant. This is the reason why scanning needs to be automated in the build process as opposed to it being an add-on activity at the end of a deployment. Tools such as Sysdig or similar can be used to continuously scan image registries as part of secure container orchestration and monitoring process.

2.2 On-premise Security Architecture

This section examines the traditional on-premises security architecture with an aim of highlighting the traditional IT security approach and how it differs with cloud native security approach in public clouds. Traditional IT security has evolved through the years in its approach, network defence techniques and tools used to meet security challenges in distributed network infrastructure. In a hybrid scenario, reconciling legacy security methodologies with public cloud security architecture is a balancing act. The architecture of classic IT Infrastructure is principally demarcated by network firewalls between an inner network and a demilitarized zone (DMZ), which interfaces with external endpoints such as internet [6]. Figure 6 shows an Illustration of a tradition network infrastructure security topology. The firewall is the primary network defence construction for traffic filtering and access control.

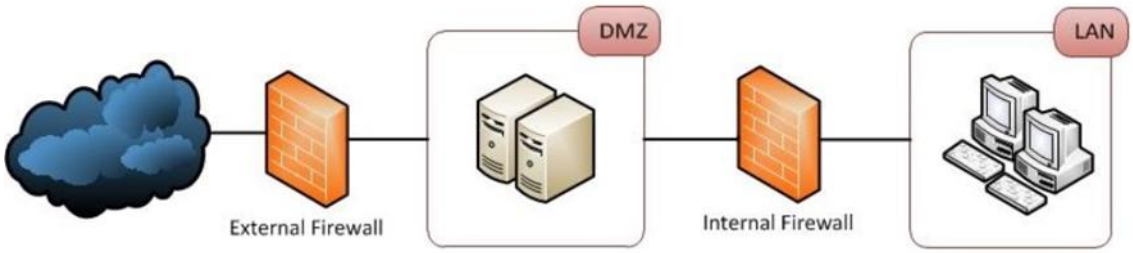

Figure 6: An illustration of a traditional perimeter network security.

The security approach depicted in Figure 6 depicts a traditional DMZ setup for connecting internal IT infrastructure to external Wide Area Networks (WAN) through a perimeter network whose primary defence is a series of firewalls. A perimeter network is defined as any network that provides services to unknown users or networks [7]. The services provided on perimeter networks includes internet access to enterprise networks. In cloud deployments, network firewalls are virtual devices or appliances provisioned as managed security services that can be used together with security groups, policies and access lists to secure virtual networks. In a hybrid deployment configured with VPN tunnelling over public internet, a topology with a virtualized perimeter network adds a layer of security for terminating the cloud to an on-premises connection. This is not always a requirement as some deployment terminate the connection into an edge gateway.

2.2.1 Demilitarized Zone (DMZ) in Cloud Infrastructure Security

A demilitarized zone (DMZ) is a perimeter network which shields an enterprise's local-area network (LAN) from untrusted traffic. DMZ works as a network security buffer which needs to be deployed with an external facing firewall used to filter traffic headed into an enterprise network from the outside [12]. The security logic is premised on identifying users and application attempting to connect from external sources. The cloud security setup can also be architected with a virtual DMZ, mimicking the perimeter network topology. The architecture of a classic or cloud DMZ is normally implemented with a subnetwork that sits between the public internet and private networks. The connection point to less trusted networks therefore terminates in a DMZ as Figure 7 depicts.

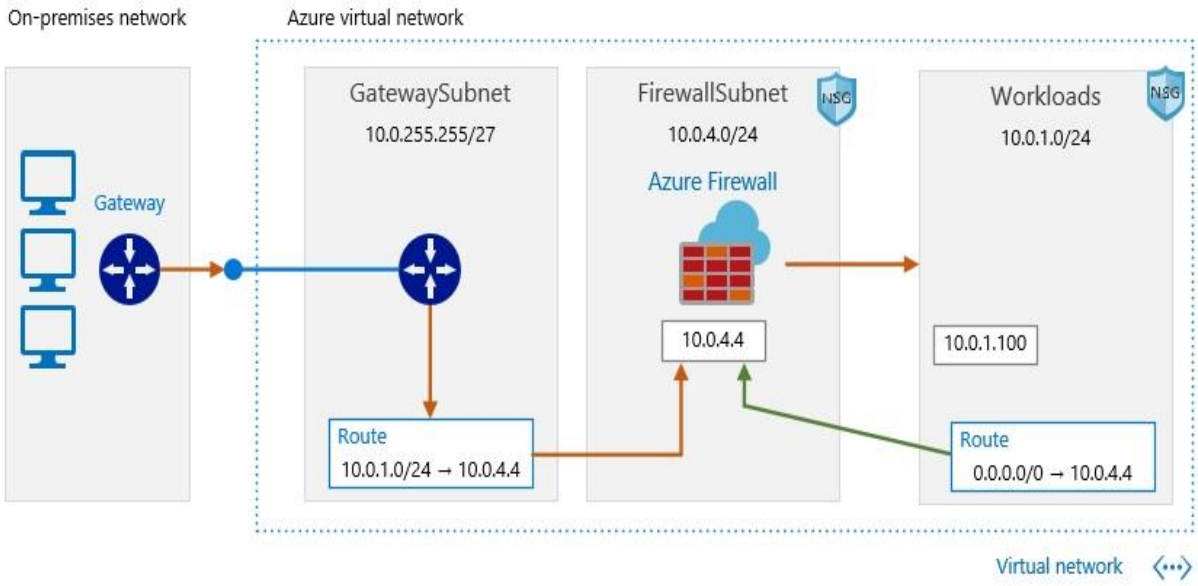

Figure 7: An illustration of a Cloud DMZ.

DNS servers, FTP servers, mail servers, proxy servers and web servers are ideal DMZ services. Installing proxy servers in the DMZ helps to simplify the monitoring of connection activities. It is important that a perimeter network is architected in a way that affords a seamless hybrid application access across the two environments without complexity. The design and architecture approaches can include single, dual or multiple firewalls. The majority of modern DMZ architectures use dual firewalls that can be expanded to develop more complex systems. Intrusion detection system (IDS) or intrusion prevention system (IPS) within a DMZ are advised in hybrid environments.

2.2.2 Hybrid Edge Security and Firewalls

Connecting a hybrid environment while retaining strict security requirements, requires addressing security also in the context of scalability of the connection bandwidth. Devices such as Virtual Private Network (VPN) gateways have bandwidth and performance limits, they may require to be provisioned as many as carrier capacity can enable, to satisfy performance requirements. A dedicated connection e.g. an ExpressRoute, as illustrated in Figure 8, will provide better security and equally provide higher bandwidth with lower latency than a VPN connection. Dedicated private connections enhance security and are preferable. A dedicate connection setup for hybrid connectivity is usually connected to edge routers on one or either sides of the cloud edge.

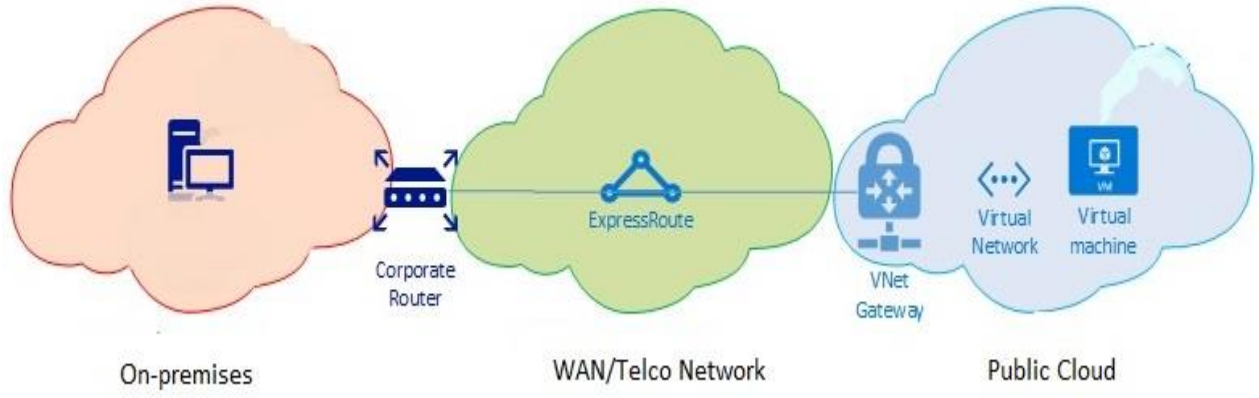

Figure 8: An illustration of a hybrid connection to a public cloud.

With public cloud computing, the network perimeter is essentially absent since users access resources directly from the internet. The challenge here then becomes determining where the security layers need to be placed and what type of security gates need to be deployed to encapsulate the deployment [19]. Cloud edge security in particular needs to be addressed when architecting hybrid environments targeted for web or Internet of Things (IoT) applications. From an edge infrastructure perspective, attention needs to be given to addressing security configurations on the cloud edge. Securing access to edge compute resources is best implemented with the aid of encrypted tunnels and application firewalls if not access controls. Secure Access Service Edge (SASE) implementation needs to be a security requirement in a hybrid architecture of an IT deployment, the working concept being the creation of a holistic WAN capability which

securely manages network security functions on the edge. A good example of the aforementioned being secure web gateways (SWG), firewalls as a service (FWaaS) and zero trust network access (ZTNA). SASE as a security concept is new yet indispensable as a hybrid solution and strategy. Figure 9 illustrates an example edge security implementation using a next-generation firewall to secure a web infrastructure edge. Web application and Internet of things (IoT) represent an area that requires intelligent edge security as they have public facing operating interfaces.

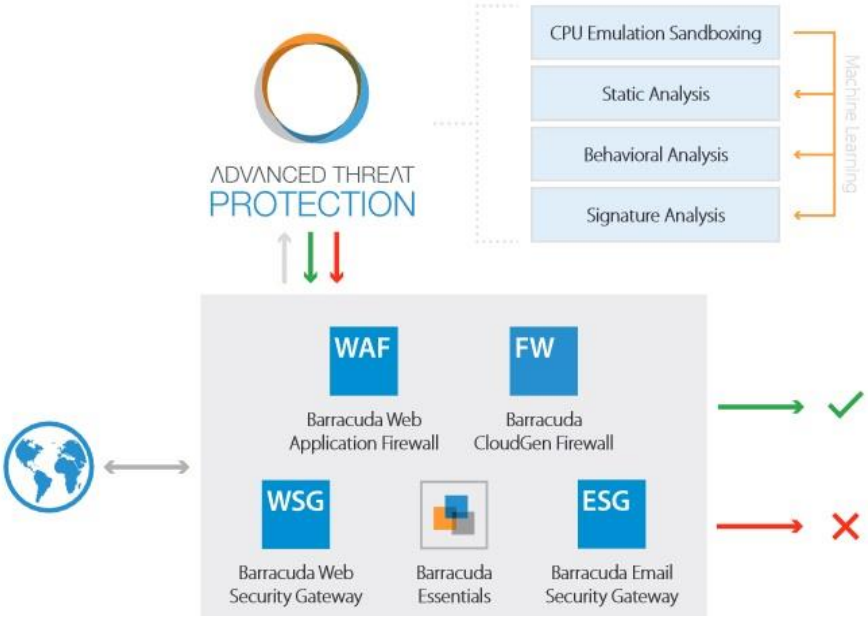

Figure 9: Overview of a (NGFW) [19]

Although modern security approaches are identity centric, as opposed to perimeter centric, next-generation firewalls (NGFW) intelligence mechanisms enable intrusion Prevention System (IPS) capabilities with Deep Packet Inspection (DPI) which enhances security by helping detect malware for example. The advantages with NGFW includes an ability to integrate threat protection technologies on the edge. For application with web interfaces, a Web Application Firewalls (WAF) or application gateways for traffic filtering can best be included in the architecture. Firewall-as-a-Service (FWaaS) or distributed third party firewalls are solution alternatives as shown in Figure 9. A next generation firewall can be used together with a WAF in a complementary security set-up. NGFW's application aware functionality will secure internal clients when accessing the internet but not internal applications from external threats which WAF's will do.

3 Current State Analysis

This chapter analyses the current state of hybrid cloud security. The research references Microsoft, Amazon, VMware, Google and Red Hat cloud offerings. The current state of hybrid cloud security can best be explained by highlighting the cyber threats and security challenges posed to hybrid computing. Evaluating the treats provides a contextual outlook of what security solutions would be required to address these threats and challenges using available tools and technologies.

3.1 Security Threats and Challenges

IT systems and infrastructure deployments will always have security threats. Cloud providers have different emphasis and approaches to mitigating threats. Table 4 below lists some of the common security threats in hybrid clouds as evaluated by the security company Pulse Secure.

Table 4: Common Hybrid Cloud Security Threats [Pulse Secure]

1	Lack of encryption	2	Inadequate Security Risk Assessment
3	Poor Compliance	4	Weak Security Management
5	Poor Data Redundancy	6	Failure to Authenticate and Identify
7	Unprotected APIs	8	Denial-of-Service (DoS) Attacks
9	Distributed Denial of Service (DDoS) attacks	10	Poor IP protection
11	Lack of Data Ownership	12	Failure to Communicate with Cloud Provider
13	Poorly Defined SLAs	14	Data leakage
15	Poorly-Defined Management Strategies	16	Badly constructed cross-platform tools
17	Disgruntled or Malicious Employees		

How to protect infrastructure, when faced with challenges such as listed in Table 2 is not defined by any single security strategy. What is common instead are security methodologies, frameworks and security patterns. The important question which arises is, “how can an enterprise mitigate such security threats against all complex challenges of hybrid cloud integration”. When architecting security for hybrid IT, it is important to develop solutions by first analyzing the threat landscape and known attack patterns.

Table 5 below shows an example approach. Here a container attack matrix is highlighted. The table illustrates exploitable techniques and weaknesses that have been observed in Kubernetes containers.

Table 5: Kubernetes attack matrix

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Impact
Using Cloud credentials	Exec into container	Backdoor container	Privileged container	Clear container logs	List K8S secrets	Access the K8S API server	Access cloud resources	Data Destruction
Compromised images in registry	bash/cmd inside container	Writable hostPath mount	Cluster-admin binding	Delete K8S events	Mount service principal	Access Kubelet API	Container service account	Resource Hijacking
Kubeconfig file	New container	Kubernetes CronJob	hostPath mount	Pod / container name similarity	Access container service account	Network mapping	Cluster internal networking	Denial of service
Application vulnerability	Application exploit (RCE)		Access cloud resources	Connect from Proxy server	Applications credentials in configuration files	Access Kubernetes dashboard	Applications credentials in configuration files	
Exposed Dashboard	SSH server running inside container					Instance Metadata API	Writable volume mounts on the host	
							Access Kubernetes dashboard	
							Access tiller endpoint	

Having dimensioned the threats, choosing a security approach for implementing a secure topology which mitigates the weaknesses highlighted after modelling a cyber-threat by analyzing the threat landscape is where requirements should be derived. Requirement definitions need to align with security benchmarks and policies. What is then required is developing a security outline of specific security measures which effectively meets the requirements. General measures in security practice includes:

- Segmentation and least privileged access
- N Factor authentication.
- Security gates (firewalls, network segments, security groups, access lists)
- Secure identities and encrypted storage
- Access rules and policies.

The solution in general needs to include as many elements of cyber security constructions as can possibly work together in a security implementation of a deployment.

Considering that public cloud security is based on a shared security model, it is critical that no security gaps are left uncovered when architecting for hybrid security. The foregoing means that enterprise security architecture should be aligned with the defined security boundaries in the responsibility domain of a public cloud consumer. It is recommended that architecting from an assumption that the provider's security measures are not completely attack proof, is the best approach which is the reason any infrastructure that can be encrypted has to be encrypted. In practice, security governance needs to be implemented using best practices that are effective for specific types of threats. Figure 10 illustrates how cyber-security threat mitigation methods have evolved in the last 2 decades. Around the early 2000's, network defense relied primarily on network firewalls, this has since chnaged to security strategies such as those described in this thesis like zero trust and identity driven security.

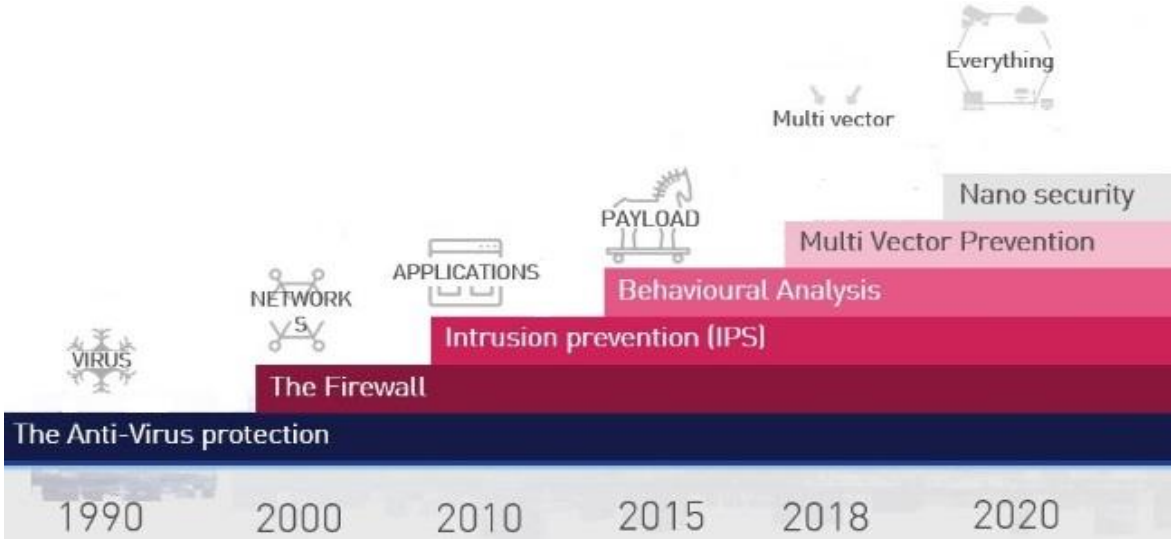

Figure 10. Cloud Security shared responsibility scope across offering, Microsoft

From an enterprise perspective, extensive security planning is required before extending traditional IT to the public cloud. Public cloud side security vulnerabilities are often cited as a leading barrier to cloud adoption [13]. Hybrid security in heterogeneous hybrid clouds can be challenging to implement. A cloud broker would be required to provide a unified API and console interface. Seamless extension of the public cloud into the local data center should be the security goal. The fortunate part is that, the IaaS layer of both private IT and public clouds use similar computing, storage, and network entities which can be protected by the same security means and technologies.

3.2 Threat Mitigation, Methods and Approaches

Having understood the security threats and challenges, mitigating threats becomes an issue of first evaluating technologies and methodologies for securing hybrid infrastructure. The architectural decision for selecting which security technology meets a deployment scenario can now be chosen according to the bench-marked security requirements. The goal at this stage is creating a security outline in respect of the defined security baseline. The baseline often contains enterprise IT security standards applied across the organization, such as how and who has access to which infrastructure or what operations on specified resources they can perform. Hybrid cloud architecting principles are cloud agnostic since security aims are similar across different cloud providers. Listed below are some security design principles advanced by Amazon in its public cloud. These principles are not any different from other providers insomuch as the approach to implementing security patterns may be slightly different from one provider to the other.

1) Implement strong identity
2) Enable traceability (Monitoring)
3) Apply security at all layers
4) Automate security best practices (IaC, DevSecOps)
5) Protect data in transit and at rest (Encryption)
6) Prepare for security events (Threat response)

The above listed security principles aid in the formulation of a security posture. Different providers place different emphasis of which principles their security approach prioritizes. The provider reference architectures are often basic security patterns requiring customizations tailored to different workloads and cloud services.

Amazon Web Service (AWS) well architected framework for example does not recommend any technologies but suggests technical configurations that ensure secure operation of cloud infrastructure. The well architected framework points to 7 pillars of cloud security according to AWS implementation and approach to security. Three security pillars stand out in advancing infrastructure security. The aim is to implement a strong identity foundation, applying security mechanisms at all layers of infrastructure and automating all security best practices or configurations [19, 13- 14] .

Building a security architecture meeting the requirement definitions is made easier when technologies that can achieve a set of security goals to meet the requirements are known and understood. Having highlighted much of the hybrid cloud security threats. A summary of 6 (six) areas of focus can be summarized as follows.

1) Configuration automation and tools
2) Security strategy and posture
3) Access management
4) Secure design for workload communication
5) Visibility
6) Implementing security as code

Using Infrastructure as code enables automating security since the infrastructure implementation in machine-parsible language or domain-specific language (DSL) can be understood and manipulated to implement security gates or apply rules and polices to infrastructure elements. Common tools for applying security with code to infrastructure include configuration management tools like Chef and Puppet. These tools and similar others help to implement security automation and create a consistentt application of security. The goal can be summarized as shown in Figure 11. Configuration management removes humans as a potential bottleneck for infrastructure security at scale.

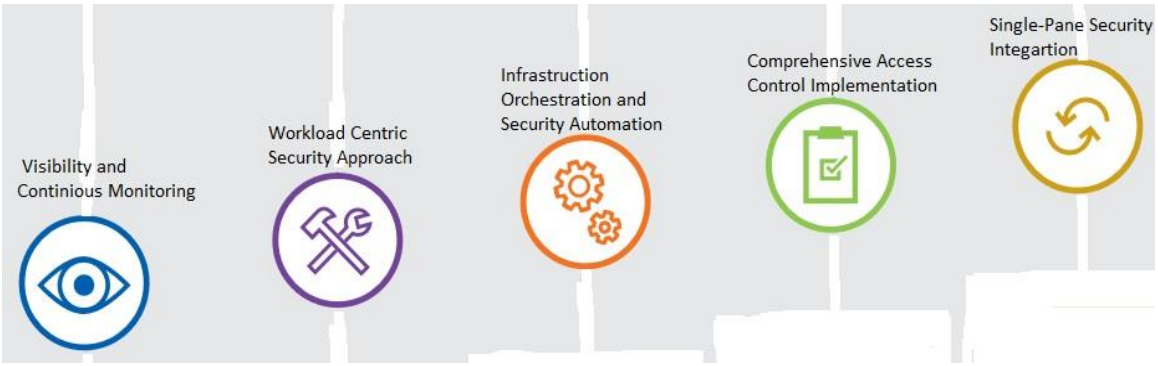

Figure 11. Hybrid Cloud Security orchestration, Microsoft

Applying security as code is efficient when applied on infrastructure described in code. Central to the above stated objective, is identity and access management. To summarize the current state of hybrid cloud security as of the year 2021, the architecture of hybrid IT is still evolving. Research firms such as Gartner and Forrester all estimate that up to 60% of enterprises will posture towards hybrid infrastructure [15, 13, and 16].

3.2.1 Securing Private to Public Infrastructure Connection

Many security principles border on common sense, an example being the fact that routing traffic over the internet exposes an organization to greater cyber risks. The connection between an on-premises environment to the cloud is the first security area of primary concern which needs to be planned when setting up hybrid infrastructure. There are not many options to choose from when deciding what form of connection to deploy. Whether to employ a dedicated private Wide Area Network (WAN) connection or encrypted tunnel over public internet through a Virtual Private Network (VPN) needs to consider factors such as; capacity, reliability, cost and data through-put. Despite the fact that an encrypted Virtual Private Network (VPN) over public internet is referenced in alternative architectures, a dedicated Open Systems Interconnection (OSI) Layer 3 connection between on-premises network and the public cloud is preferably secure and reliable way of integrating hybrid IT as illustrated in Figure 12.

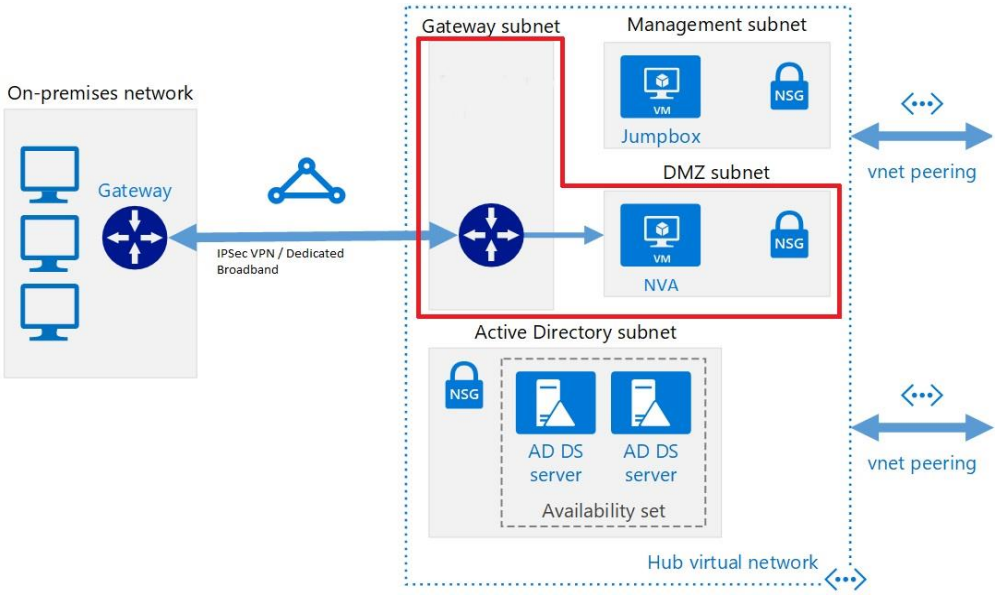

Figure 12: Dedicated connection between on-premises and the cloud

The principal security attributes for a dedicated connection between the private and public environment is based on encrypted tunneling either over a dedicated broadband link or an IPsec virtual private network (VPN) connection over public internet. It is sufficient to terminate dedicated connection into a VPC/VNET without a DMZ edge network as shown in Figure 12. It is logical to build a connection that supports private addressing (RFC 1918) so that the cloud extension of on-premises infrastructure requires no public IP addresses or Network Address Translation (NAT).

3.2.2 Cross Environment Infrastructure Security Configuration

Security infrastructure and applications that span both private and public infrastructure is another hybrid security challenges whose design considerations is central in realizing hybrid IT security. For distributed computing deployments, the primary infrastructure is the network, which has to be hardened where access to infrastructure is concerned. A cloud DMZ setup can enhance security provided it does not add bottle-necks to transiting traffic. Some security architectures see no need of having a firewall against traffic coming in through a private connection especially in a zero-trust network where identity is the new perimeter and not firewalls. It is however highly important in conforming to layered security approach that a topology uses edge firewalls and a DMZ. Figure 13 shows an alternative secure architecture which uses a router in a gateway subnet .

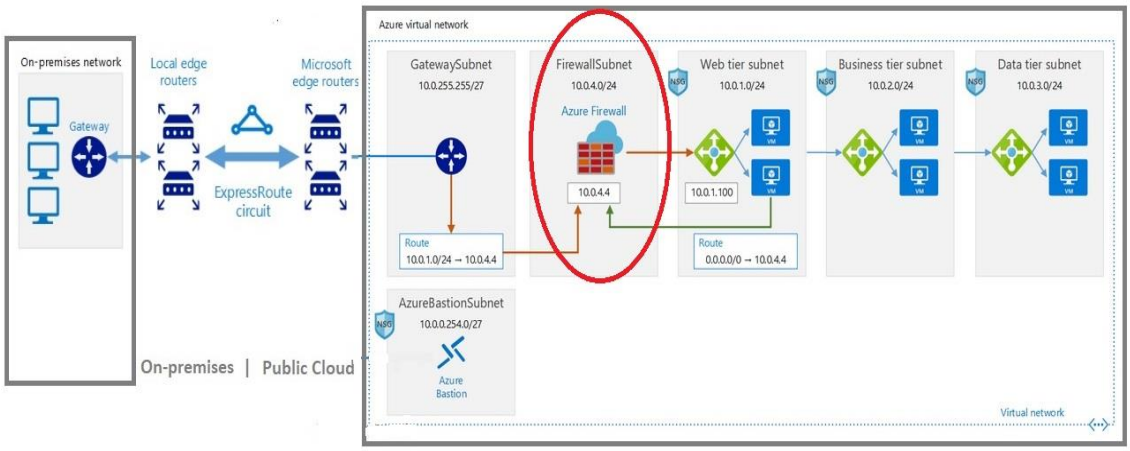

Figure 13: Routing through firewall with dedicated private connection

Amazons public cloud approaches the architecting of hybrid security with emphasis on five security pillars among them being encryption, data redundancy and security management. Operating a hybrid-cloud requires a security centered approach above all else. Insofar as they are many managed security services offered in public clouds, not all can be part of the security baseline as much as they can serve as security addons. Having a reference architecture based on properly defined security baseline requirements saves on security costs by not enabling features that duplicate functionalities within a security architecture. Edge security in hybrid architectures is becoming complex with connected IoT devices which introduce an increased security attack surface. IoT security is not discussed in this thesis, suffice to say it is part of a broader edge security strategy, from a general point of security solutions described in this thesis.

3.3 Security Architecture with Security as Code (SaC)

This section highlights architecting security as code (SaC) solutions using DevSecOps. Implementing security as code can is one of the best approaches for an enterprise scale deployment security orchestration. Code and automation are two sides of the same coin and hence the certain benefit of eliminating security misconfiguration by implementing tested security constructions. Infrastructure misconfiguration whether by manually built implementations or code driven deployments can get complex and too large to easily manage. The idea behind (SaC) is to consider security a part of infrastructure life cycle. What is achieved is describing and embedding code to be orchestrated concurrently with infrastructure during the deployment process. Figure 14 below shows the principle behind the idea of orchestrating security as testable code.

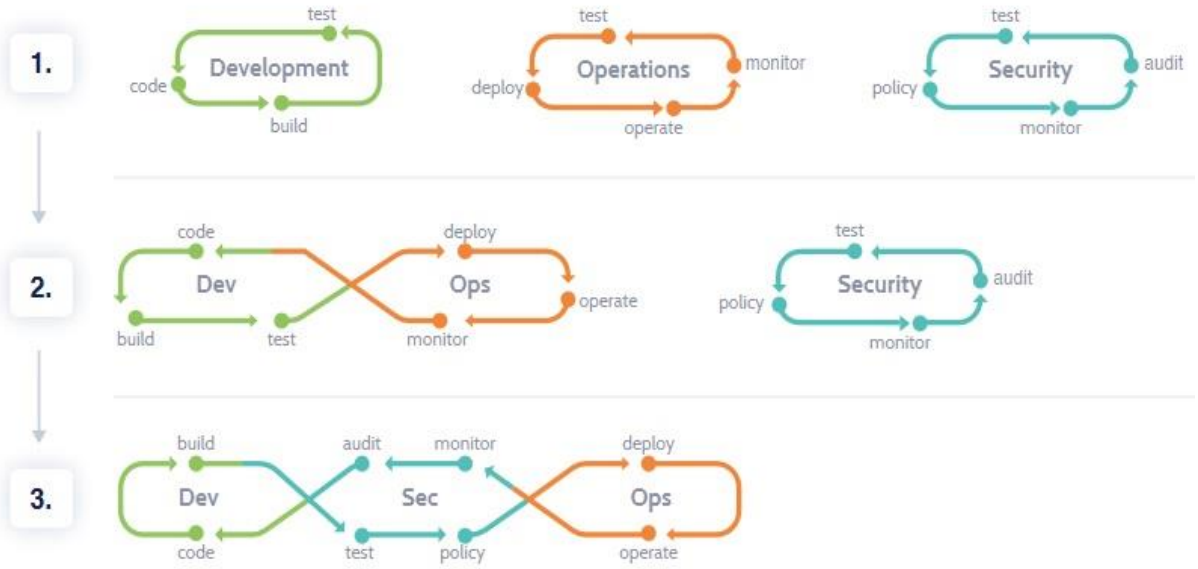

Figure 14: Delivering Security as code by DevSecOps

The solution approach to the challenge of making security a part of infrastructure life cycle by design requires configuration, orchestration and a secure management that can be updated in a modular way or by functionality with code. Modularized infrastructure components are easier to secure, test, update and quickly troubleshoot. It is also hard to lose track of what is happening when components are modular. Using secure DevSecOps as illustrated offers the best approach to deploying secure infrastructure as code. The process in Figure 14 when applied to infrastructure security sets up a process of being able to either continuously introduce or test security implementations.

Security logs and SIEM can be leveraged as a source of actionable insight by being integrated into a hybrid security implementation. The resulting security design of SaC with an input and feedback process to remediate possible security loose ends is made possible with configuration management automation or desired state configuration (DSC). As an example, whenever new infrastructure is deployed, security tests need to be run against the deployment. Figure 15 illustrates the use of Ansible Tower as an automation engine to orchestrate and managing security governance.

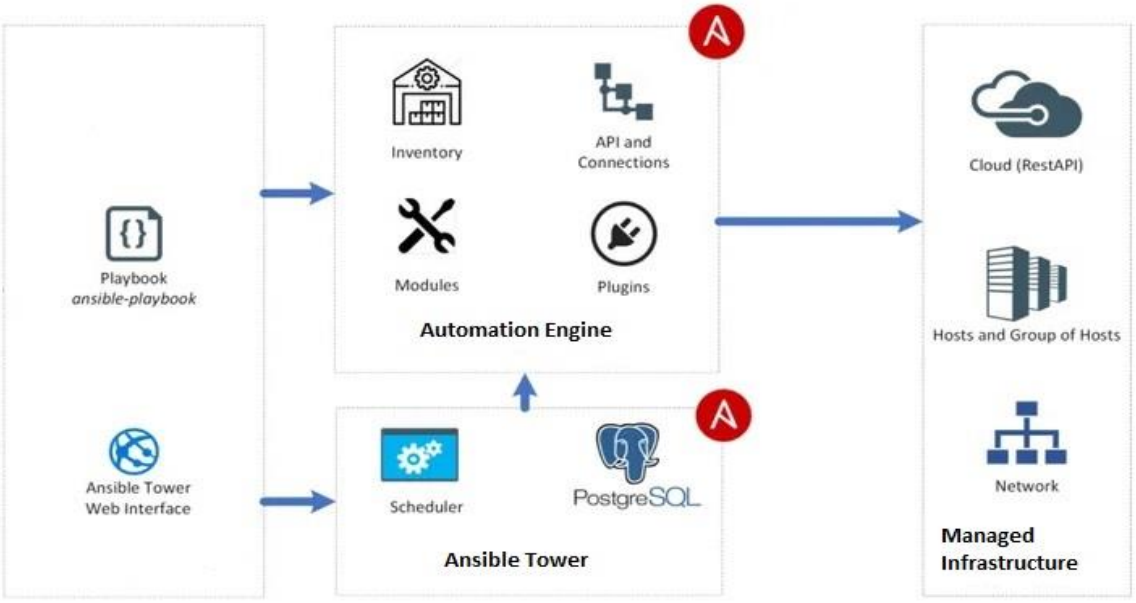

Figure 15: Ansible Automation

Figure 15 highlights the point that iinfrastructure security automation requires a configuration orchestration tool which Ansible tower or similar tools represent. Orchestration tools are capable of offering the listed benefits in implementing security by code:

- Orchestration from a central location
- Configurable with easily updated code
- Leverage DSC descriptive playbooks (YAML etc.)

Security automation can in practice be orchestrated with the same infrastructure automation tools as those used for infrastructure as code tools. For example, network security devices such as F5, Check Point, Cisco which are used as firewall solutions can be manages with any code-based tool to apply DSC integrated with monitoring.

The aim of security automation is best summarized by Figure 16 below which illustrates a high-level security management flow for remediating security incidents. The architecture includes an anomaly detection systemm and response orchestrator. The anomaly detection will mine the security alert logs and trigger actions to remediate the threat like for example updating firewall rules or prompting further authentication.

Figure 16: Simple Security Workflow for security response

The abilities to implement a continuous intelligent analysis of the security integrity of an infrastructure needs to be structured in a lean manner as to provide greater visibility and less complexity. Zero trust network security employs micro-segmentation as a security strategy [13]. Micro-segmentation is the practice of breaking up security perimeters into small zones to maintain separate access for separate parts of the network. Figure 17 illustrates a high-level overview implementation of zero-trust.

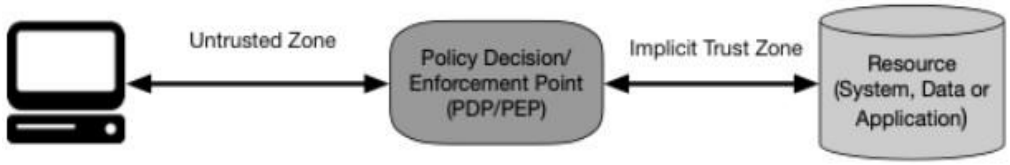

Figure 17: Steps for implementing Zero trust [Duo Security]

A network with resources situated in a single data center or network can be segmented into separate security zones to limit security damage in any case of a successful breach. A user or program with access to any protected and segmented zones is not able to access any of the other zones without separate authorization. This is the security concept behind segmentation which advised in hybrid implementations. In the cloud this is achieved on management, subscription, resource or security group etcetera.

3.4 Evaluation of hybrid reference architectures

Evaluating some common hybrid security architecture patterns is important for an in-depth understanding of why given architectural decision are important or why they need to be part of security requirements. Figure 18 below shows 3 important security requirement considerations being:

- Secure cross cloud interconnection achieved by any of the following options: encrypted VPN or dedicated WAN connection.
- Leveraging provider data center backend network for infrastructure sitting on the same data center through VPC/VNET peering.
- Prevent cross environment workload communication.

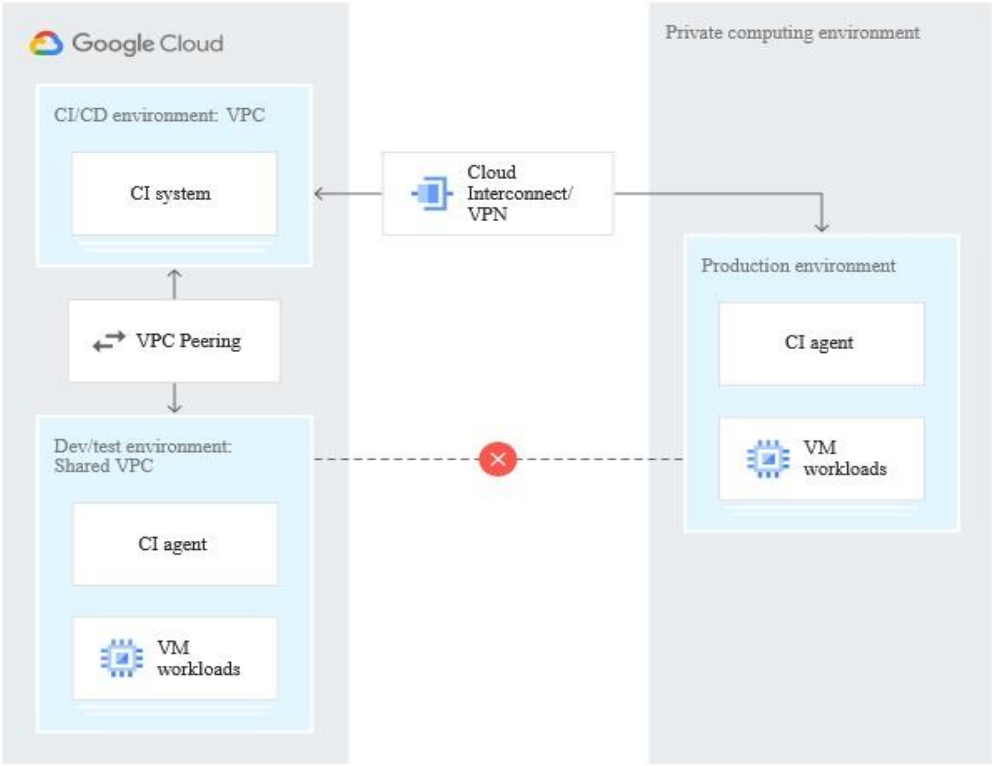

Figure 18: Enabling cross Environment Integration across a Hybrid Tapology

Figure 18 illustrates how to isolate hybrid infrastructure across environments as a security architecture. This important approach is a best security practice solution and is advised as a security requirement. Isolating different environments such as development, staging and production etcetera is also possible by either provisioning separate management groups, subscriptions, VPC's prohibiting interconnections.

Virtual Machines (VM)s dominate the compute infrastructure in hybrid IT as lift and shift migration of legacy IT systems normally means deploying servers by direct translation to a virtualized (VM) equivalent of a physical server. Securing virtual server infrastructure in a hybrid environment can be implemented by avoiding public IP addresses that connect directly to the internet. Using Network Address Translation (NAT) on the edge to handle egress traffic as illustrated in Figure 19 is instead advised. The Idea is that private IP address should be used across provisioned infrastructure locally.

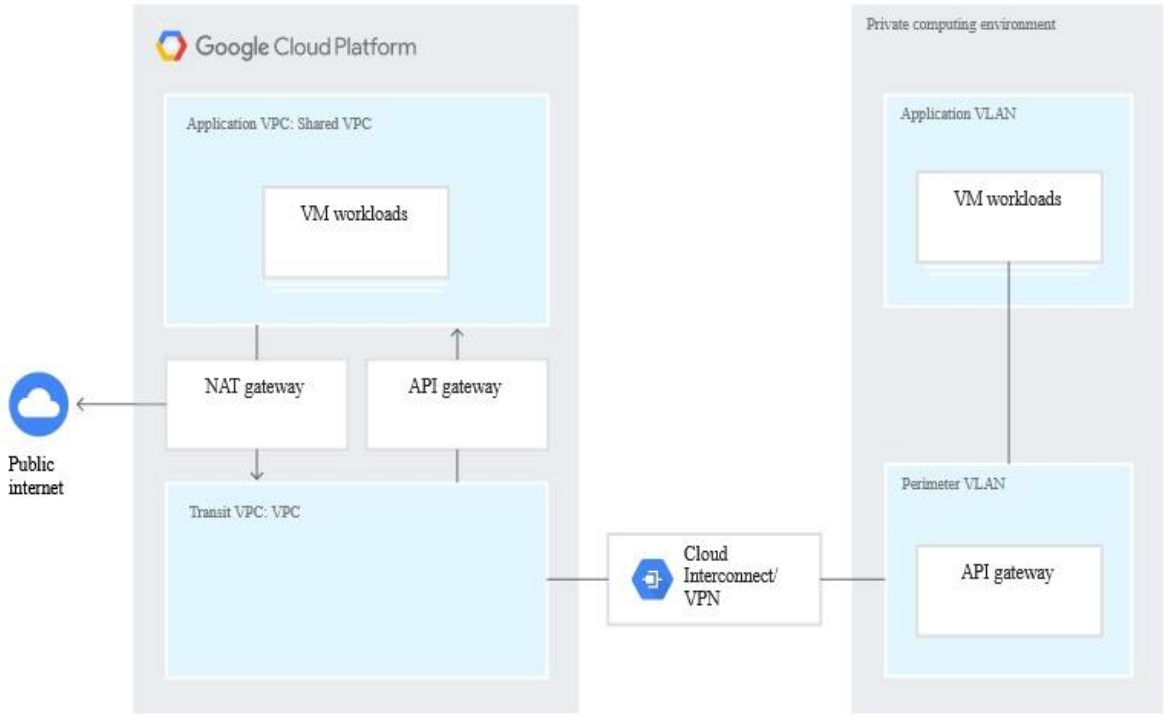

Figure 19: API Gateway as NAT for internet facing VPCs

Some other methods of encapsulating private IPs is to use API getaways to manage any transformation between API protocols and control access through security schemes or quotas. Applying a gated-egress security implementation is recommended for edge hybrid compute scenarios where APIs communication across the back-ends of tiered infrastructure which should not be exposed to the internet. To enable bi-directional usage of an API across workloads requires gated ingress and egress on both sides or connected VPCs or environments. The security benefits of using APIs includes the enforcing of authentication and authorization of calls (communication). NAT and API gateways provide a means of securely enabling communication to sources external to an infrastructure deployment or internal workload communication restrictively. Security rules and white lists can be attached in traffic filtering here.

Firewalls in IT infrastructure are indispensable in implementing security. In hybrid deployments the need for layer 3 or Layer 4 firewalls between the public internet and a cloud network can enhance security by filtering traffic based on the source or target infrastructure. Different firewalls provide different capabilities. Whether to use a software-as-a-service (SaaS) firewall, security as a service (SeCaaS) or firewall as a service (FWaaS) depends on a use case. A stateful firewall is ideal in hybrid topologies to constantly analyzing the complete context of traffic and data packets [ ]. Dynamic packet filtering will help monitors the state of active connections for determining which network packets to allow. Figure 20 illustrates the positioning of an L4 internet facing firewall. This is more flexible than static packet filtering.

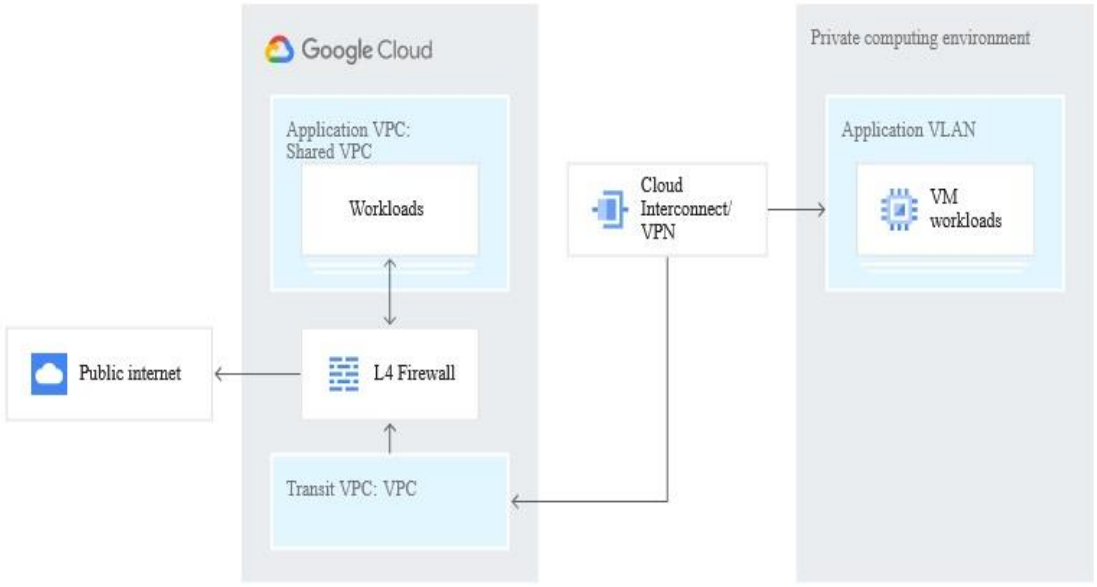

Figure 20: Lawyer 3 and 4 internet facing packet filtering.

When considering stricter isolation and control, implementing deep packet inspection with advanced firewalling needs should be enabled for traffic across either isolated environments or towards the internet. Firewall appliances between the transit VPC and the next VPC can be used. The firewall appliances can also be used for IP forwarding or serve as a NAT gateway for internet facing infrastructure deployed in an isolated VPC with private IP address. This type of approach limits IP address-based security exposure. API's can secured easily to connect resources and workloads. Having an API gateway inside a perimeter network (DMZ) will secure workloads in private internal virtual networks while employing non routable IP addresses internally.

3.5 Container Security Architecture in Hybrid Deployments

Hybrid compute security currently is not complete without addressing container technology. Container security is built around container management components like Kubectl, container registries and images. Comprehensive container security is not possible to explain within the scope of this thesis and is here only discussed briefly at a high level. The general approach to container security from an architectural perspective in hybrid environments when architecting for deploying container services can be enabled by securing the components at the level labelled A-D in Figure 21, in addition to implementing secure access controls as shown in the same figure.

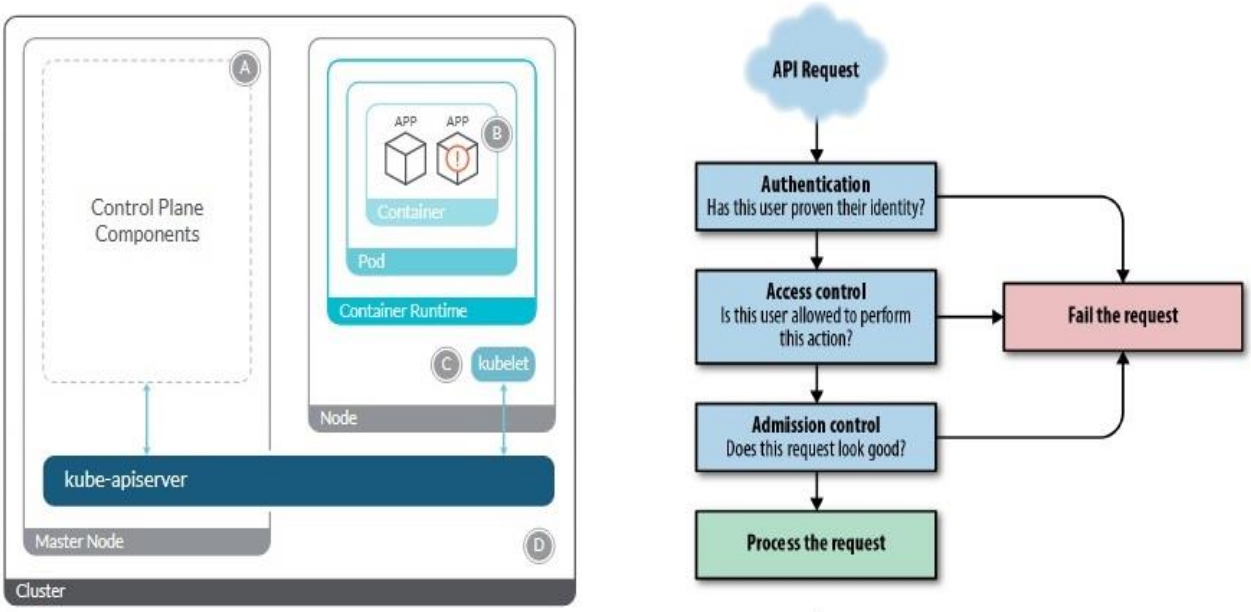

Figure 21: Container (Kebernates) Architecture and Authentication

Containers being immutable infrastructure require a different security approach but with the same context of a broad security strategy as described in this thesis. There are many security measures that can be employed for authenticating users, connections and applications in container deployments. Basic authentication, X.509 client certificates or Bearer tokens. The manner in which a user ultimately authenticate depends on the identity provider and installed authentication system in an enterprise. Although authentication mechanisms are vastly different in terms of how they are implemented. The API server on the control plane needs to implement verification and authentication ensuring that not only are user-initiated requests transmitted securely, but also that service-to service communication is encrypted with X.509 client certificates or secrets.

Covering container infrastructure security may require a study of its own. The aim in the present study was to only contextualize its implementation in a hybrid setup. For purposes of preparing the implemention of hybrid cloud compute infrastructure, securing container components should leverage the same set of tools and security philosophy. Figure 22 provides an overview of the management elements of the containers. Containers can be secured with a gated topology for enforcing stricter isolation. Admission control in containers like other cloud infrastructure resources can be in granular way using Role Based Access controls (RBAC).

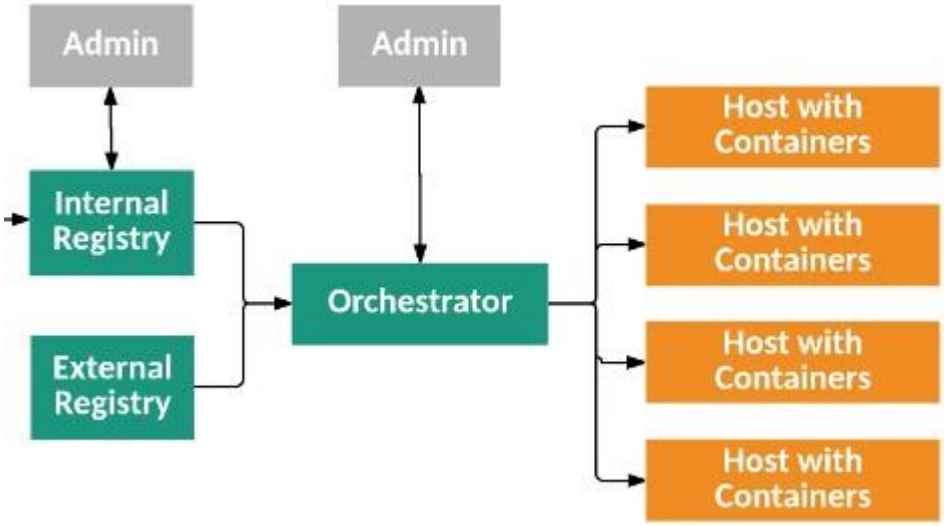

Figure 22: Container Infrastructure topology

As with all software, containerized applications have system security vulnerabilities of various kinds, including bugs, inadequate authentication etcetera adding on to common issues of misconfiguration. Container architecture itself being made up component that require different security configuration complicate properly securing them. National Institute of Standards and Technology's (NIST) has pointed out some container vulnerabilities such outdated runtime engines such as containerd, CRI-O, and rkt. It is important to keep runtime security patches up to date. NIST also recommends running a bare-metal container-specific operating system.

4 Hybrid Cloud Security Solutions

This chapter provides a technical approach of implementing secure hybrid solutions. The aim is to determine reliable means of implementing hybrid security at scale.

4.1 Approach to Network Security in Hybrid Environments

The approach for securing networks is here presented as a set of recommendations. The recommendations aim at aachieving a low-risk network penetration. Table 6 tabulates important considerations for driving a hybrid security implementation at scale.

Table 6: An Approach to Hybrid Network Security Implementation at Scale.

Implementation Driver	**Description of Benefit**
Infrastructure and Security as Code	IaC should be used in network construction (artifacts, secrets, and configuration). They should be saved in source-code repositories.
Automate	Automation with Network Operations (NetOps) in building and network integration of infrastructure deployments. This results in security reliability, scale, efficiency, optimizations and management of dynamic provisioning of networking resources.
Verification	Testing network security components in code ensures that the deliveries are reliable and security benchmarks are verified to have been met before deployment by stages (staging, development, production).
Monitoring	Automatically process events and respond to security alerts and anomalies using logs and telemetry collected from monitoring.

As tabulated in Table 6, network security at scale is only possible with automation, continuous verification and monitoring. Orchestrating and controlling infrastructure security can only best be managed with easy when security is described and orchestrated by code. Achieving network security goals depends on controlling access to network resources. What can be summarized is that, security posturing and defined what is included in a security process; (Build, automate,verify,monitor/manage) .

The underlisted principals should always be considered in security implementations of network security as very important measures to build security around.

- Implementing firewalls and applying access rules for authorized usage
- Network access control to prevent the lateral spread of attack vectors.
- Micro-segmentation to addresses the problem of privilege escalation.

4.1.1 Cloud Side DMZ in Hub and Spoke Topology

How to setup a network topology that leverages security principles discussed in preceding sections becomes the question in need of answering. Maintaining segregation between internal and external network by having access control policies for each domain enforced is best implemented in a DMZ. This security concept has already been discussed in chapter 2. For physical segmentation, a hub and spoke topology will meet most architectural use cases to eliminate network-wide security exposure. The goal is to prevent lateral movement across an entire network infrastructure. Figure 23 illustrates an example implementation of this hybrid security concepts mentioned here.

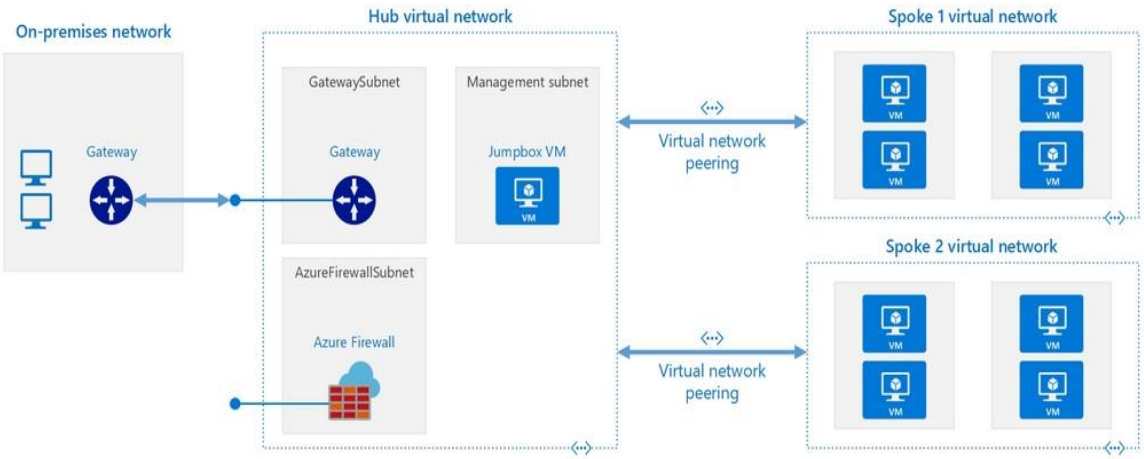

Figure 23: Spoke and hub network topology with cloud DMZ [Microsoft]

What needs to be achieved is having networks, logically and physically segmented in subnets. Traffic into the subnets can then be controlled based on functional requirements. The enhanced network topology with a cloud DMZ offers an extra layer of security. A Hub network reduces on keeping separate configuration of shared services for each segment by providing centralized management for common network services. On the cloud side of the network, using the fabric network by peering subnets keeps network connection within a provider's data center network which is insulated.

4.1.2 Software Defined Network Security

Enterprise networking has fast evolved to become software defined as opposed to traditional physical networks. Software Defined Network (SDN) as a hybrid solution provides an ability to enable secure hybrid cloud network management with global visibility of the network state around which automated security can be built [20]. Figure 24 shows the SDN architecture whose security advantage is premised on a decoupled transport and control layers as a security feature.

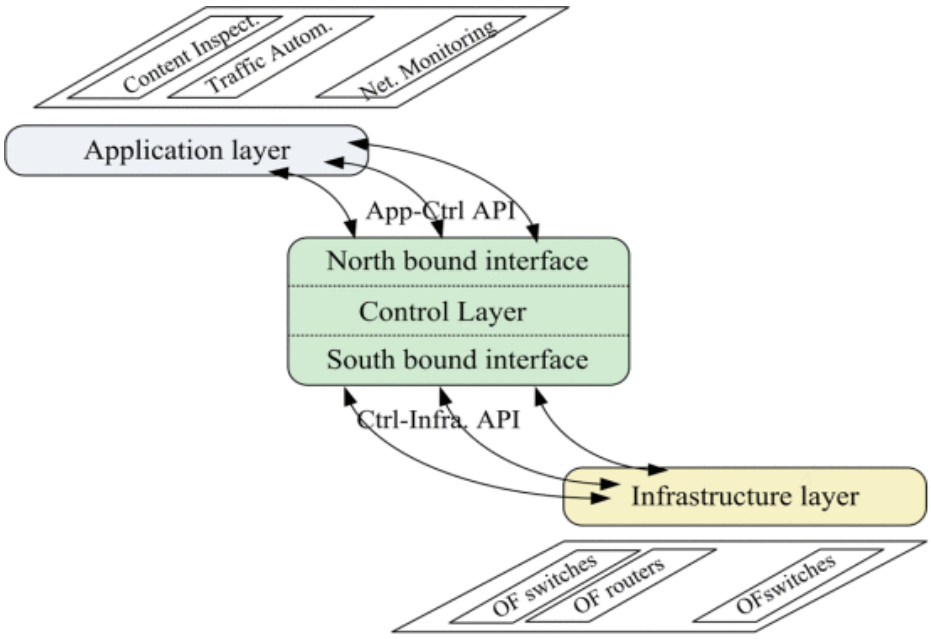

Figure 24: SDN architecture.

In hybrid IT, SDN solves many network provisioning challenges as well as improving security. The standardized ITU-T security recommendations for SDN addresses critical vulnerabilities such as access control, authentication, non-repudiation, confidentiality, traffic security, privacy, availability and data integrity as was tabulated in Table 6. Hybrid IT is not static and hence the security benefit of building infrastructure which is globally transparent, dynamic and responsive. Visibility as mentioned in many sections of this thesis, is one of the primary elements that enhances security across a platform that supports rapid change like SDN. The idea of a dynamic networking capability fits into the security as code security approach to building modern hybrid infrastructure.

4.2 Consistent Security Configurations

Archiving consistency in hybrid security configuration across environments is a challenge. The solution to solving the inconsistencies in security configurations includes some of the suggested approaches such as infrastructure as code, as an enabler for easy application of security as code. Configuration automation can be implemented as illustrated in Figure 25 with examples of Chef, Ansible and Puppet for DSC.

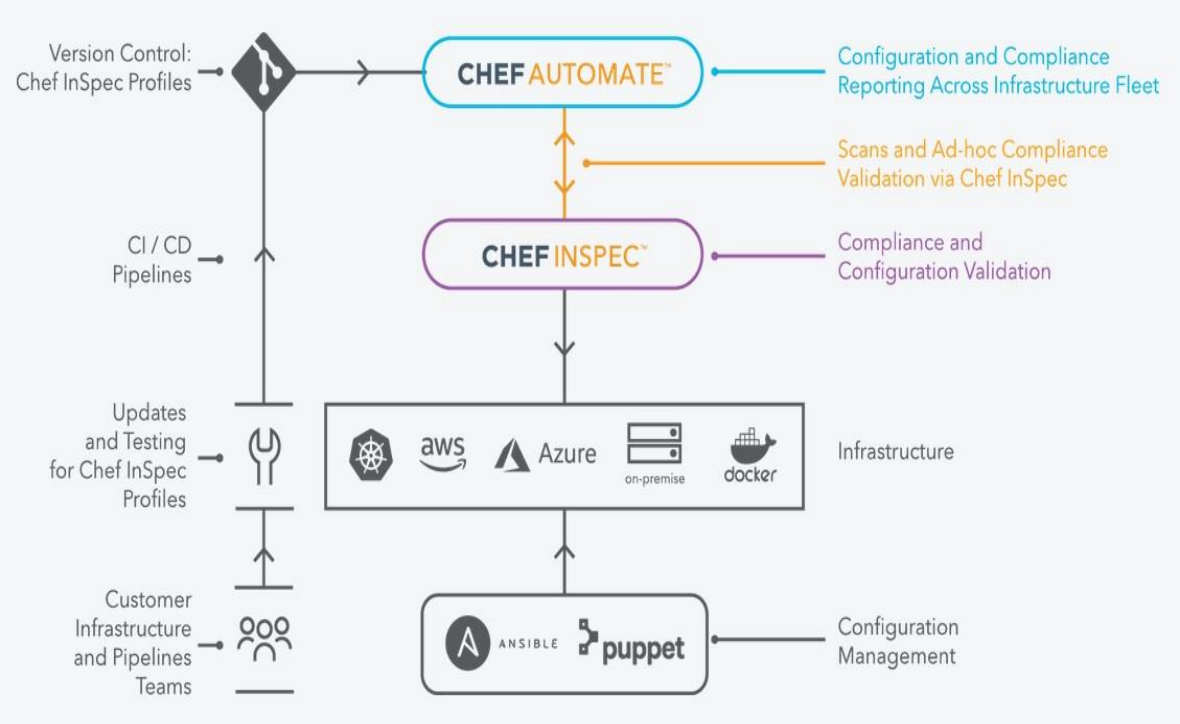

Figure 25: Security automation configuration

Figure 25 illustrates how configuration management can be implemented as part of security automation in hybrid IT. Tools like Chef, Ansible and Puppet are common environment agonistic security configuration and management tools. The agility of cloud security solutions cannot be separated from automation. Automation is the remedy for misconfiguration and security validation. The benefit and relevance of automating security includes the ability to verify whether security configurations and policies are operationally in order across the hybrid infrastructure boundaries. Adopting systems configuration tools and methods such as Ansible, Chef, and Puppet or similar script-based configuration tools is highly recommended for achieving security automation.

4.3 Centralizing Security Management

In traditional enterprise environments, security is usually decentralized, managed and maintained from different systems which monitor a segment of infrastructure or applications running on them. The question in hybrid environments arises, whether centralized security management is reliability or benefit in respect of automating security. The architectural goals for hybrid IT as described in this thesis includes reducing complexity and enhancing visibility in all corners of an environment. Centralized security management has its benefits and downsides [13]. Often security policy application can be applied evenly and consistently from a central system. Centralized security management requires identifying management tools that provides multiple features to aggregate security information across infrastructure boundaries as illustrated in Figure 26.

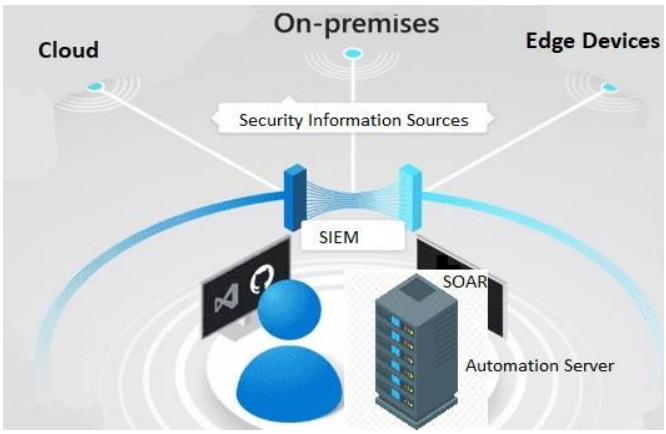

Figure 26: Centralized Security Management with SIEM

Visibility has been stated as a primary attribute for a well architected enterprise management of an IT environment. Security Operation Centres (SOC) naturally becomes an answer in practice to bringing monitoring and response operations together. There is no single right way of centralizing security or what elements comprises the operational set up. What is required to effectively respond and remediate security threats, is creating an integrated solution that aggregates information from multiple infrastructure security endpoints. SIEM solutions provide this answer for hybrid IT. Solutions such Azure Arc, McAfee ePolicy Orchestrator and Firewall Managers can be integrated in what is generalized as a security centre. Security centres often integrate infrastructure monitoring systems, threat analytics and logging systems. The ultimate security ecosystem will require integrating a response system that is both automated and human actionable for administrator actions.

4.4 Hybrid Identities

This section introduces the concept of hybrid identities. Hybrid identities are a new form of identity that can be extended to both applications and devices in hybrid clouds. They solve many problems common with credential based access control to infrastructure resources. The challenge in hybrid IT is often one of operationally different environments whose identity providers are not often the same. On-premises infrastructure normally has been built around legacy authentication providers like Windows Active Directory (AD) being the most common identity provider. Hybrid IT introduces complexity and challenges in architecting access methods that work both in the cloud and on-premises. Employing modern authentication methods enhances security across a hybrid environment. Hybrid identities are extensible and can be used in innovate ways in a security sense, as is the case with attribute based security. The novel concept of attribute based security extends the parametric nature of an identity and its complexity. This fits into the code approach to security orchestration. Code listing 3 shows an example token returned by an authentication call to an API that supports modern authentication with a hybrid identity, in its simplest representation. Extensibility means that a hybrid identity can be built as a compound object containing access information which can include many complex access variables such as certificates and hashes.

Listing 3: JSON representation of an access token of hybrid identity

```
{
  "access_token": "ekpJ….MoQ",
  "expires_in": 86000,
  "scope": "openid offline_access",
  "id_token": "eySc...0NE",
  "token_type": "Bearer"
}
```

Hybrid IT requires implementing modern identity methods of authentication. Architecting infrastructure access that leverages hybrid identities is recommended. Hybrid identities are standardized by NIST and drive modern authentication methods used in cloud systems and offerings such as Azure’s cloud based AD.

5 Security Intelligence and Automation

This section addresses security automation in hybrid IT. Cloud security research firms Forrester and Gartner both cite misconfiguration as accounting for well over 3% of public cloud data exposure [9]. The solution to addressing misconfiguration of infrastructure in hybrid environments is through a process of security automation and orchestration. Table 6 shows some security issues and there possible means of mitigation. Both architecture and security policies are important in achieving a robust security posture in hybrid environments. Table 7 tabulates some of the important security issue to which solutions are highlighted in summary.

Table 7: Selected Methods of Mitigating Security Issues in Hybrid Environments

Security issue	**Mitigation**
Access Control	ACL, Policies
Authentication	RBAC, AC
Non-Repudiation	Identities (e.g. LISP or HIP), packet validation
Confidentiality	Random host mutation, Flow rule-legitimacy, Identity-based encryption
Traffic Security	Controller switch configuration and encryption
Data integrity	network Isolation, IPsec encapsulation
Availability	Distributed Control plane

As can be seen in table 6, common security issues have varied methods of mitigation. In architecting security solutions for a hybrid environment, a decision of what can be automated and with what tools has to be determined correctly. With advances in AI and analytics, enterprise security postures have been slowly moving towards integrating AI driven security engines that can complement security automation. The common architecture for hybrid environments can include cloud provider hosted threat detection and protection engines. APIs can be used to build automated remediation that extend the built-in platform functionalities of many cloud embedded security tools. The best strategy would be constructing a security response implementation by injecting information from a monitoring platform into an orchestration engine which applies security actions whether built-in or custom scripted. Lessons from traditional security posturing attacks are only discovered after they have already occurred. The aforementioned makes threat analytics and machine learning in security automation a recommended modern approach to security especially in hybrid environments.

5.1 Orchestrating Security as Continuous Process

Infrastructure delivery tools such as DevOps can help automate security. Just like infrastructure has become attractive to be defined as code (IaC), security can be orchestrated with a similar approach. The basic setup involves combining a security feedback system which acts on security loggings of either misconfigurations or suspected threats. Security incident response platforms can be used to orchestrate security remediation by pulling and pushing code-based on threat scenarios. Figure 27 shows an example setup of a continuous integration workflow whose logical configuration can be used in the automatic remediation of SOAR. Insights obtained from mining security logs are integrated in the security process workflows that push security actions.

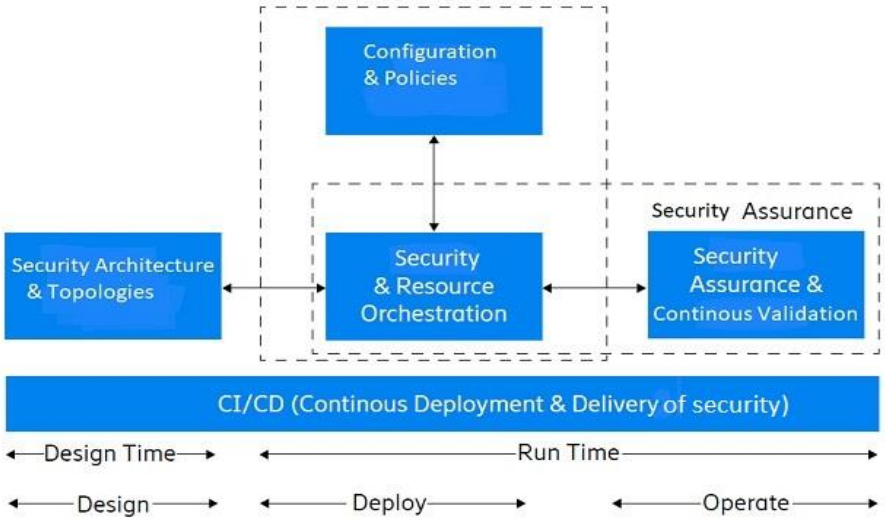

Figure 27: An example set up of a continuous integration workflow

The security idea in modern networks promotes continuous testing. Infrastructure security tests have to be part of the provisioning process and later on continuous penetration testing and policies enforcement integrity. Performance of firewalls and access controls between cloud security zones need to be in the scope of continuous security assessment and assurance. Security management with automation has both orchestration benefits and remediating compliance failures if and when they occur. On the public side of infrastructure many cloud providers enable machine learning based security. Threat analytics does not automatically remediate security issues, this is where automation becomes necessary to dynamically respond to security incidents and apply remediation using scripted security actions.

5.2 Infrastructure Security Integrity Testing

Security testing is a vital yet challenging infrastructure deployment process. Testing as highlighted in Section 5.1 is the only way of verifying and validating that infrastructure configurations have met the security baseline requirements. However security tests can only be as good as how comprehensive the security architecture is. The aim of testing is uncovering misconfigurations. There exists a logical limitation in security testing. No security test guarantees an exploit proof infrastructure. As illustrated in Figure 28, security needs to be an active continuous process enabled by continuous process of orchestration, monitoring and managing.

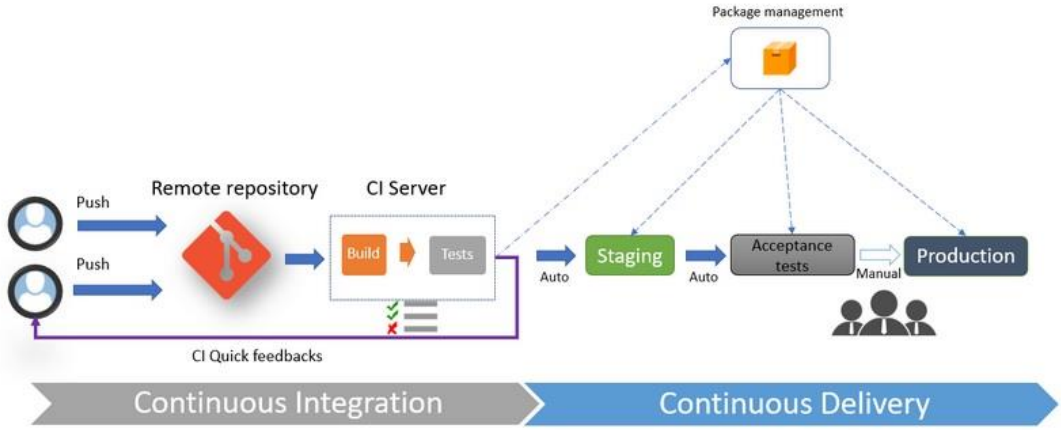

Figure 28: Security automation with DevOps Workflows

The way to develop continuous testing would require the implementation of test suits that run through the security benchmarks designed to secure a deployment. If and when the tests do not comprehensively cover all the critical security end points, it may leave security gaps and leave security weakness which can be exploited. Specific penetration test can be scripted as test cases and run against the infrastructure code. Actual security requirements tested depend on the security implemented through the selected security architecture. Adoption of continuous testing methodologies in code driven infrastructure deployments through DevSecOps improves security in hybrid environments. Benefits mostly depend on simplified automated configuration of an end to end security process.

6 Security Management in Hybrid Environments

This chapter describes how to approach security management when architecting for hybrid IT. It was mentioned in Chapter 2, that complexity needs to be avoided as an architecting principle. From the above understanding, it should be clear that infrastructure security management is a critical aspect of an end to end infrastructure life cycle. Figure 29 depicts the 3 pillars of hybrid security management.

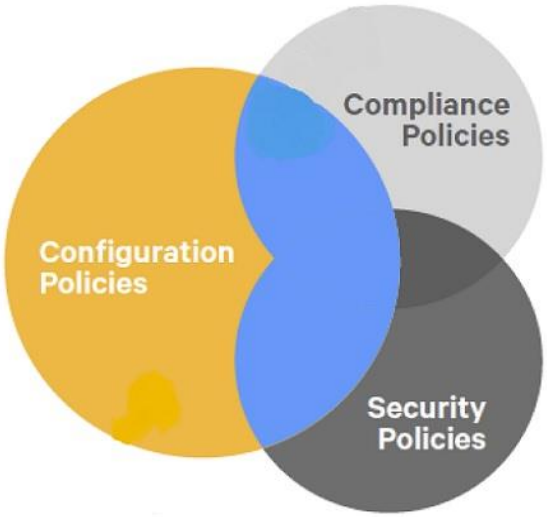

Figure 29: The 3 pillars of hybrid security management

The consequences of having a flawed hybrid IT environment whose security architectural principles are dominated by complexities of shadow IT can be avoided by a well architected infrastructure security plan. Chapter 3 covered much of the obvious integration challenges in the scope of infrastructure management. The tools used for enabling infrastructure as code such as Ansible, have been explained and how they can also be used for security management of security as code. Descriptive mark-up languages like YAML and Ansible playbooks are useful to describe security configurations. The configurations can be stored in a secure encrypted repository where they can be accessed and updated as part of the security pipeline. One effective way to approach security management is implementing it as a DevOps process than an activity. An important aspect of the said process that has to be considered carefully is the planning and architecting of security components. It is important to select the correct technologies especially one that support automation as a means of orchestration and hybrid management. Code based security management results in benefits such as flexibility to update configurations both on the private and public side of the infrastructure.

6.1 Security Management Tools and Methods

The challenges of managing security in a Hybrid environments have been highlighted at length in this thesis. Resources deployed on legacy platforms cannot easily be managed with modern tools. Morden security innovations many times conflict with legacy systems and require installing agents or being exempted from new security implementations like modern authentication. In legacy enterprise infrastructure, management tools are hosted on-premises in traditional IT date centres. The aforementioned raises the question whether tools and methodologies designed for traditional IT management can be extended to the cloud or the opposite. The answer to the above question weighs heavily on the latter being true. Tools and hybrid security methods can simplify the management of infrastructure in the listed following ways:

- Security automation
- Implementing security policies
- Security management
- Identities management

The benefit of using tools that can manage both cloud and on-premises environments cannot be over emphasized considering how much development is being put in delivering such automation and DevOps platforms such as terraform or Azure DevOps. Ideal automation tools in a hybrid environment platform have to be agonistic so as to have the same tools work across different clouds and environments. This helps with consistency in security management strategy of hybrid or multi-cloud environments [14]. Common tools that fit into this strategy includes Ansible, Chef, Puppet, and Salt, for configuration management and CI/CD like Azure DevOps or Jenkins.

6.1.1 Automating Security

The problems of security misconfiguration and complexities have been highlighted in the previous sections. Considering a manual approaches to managing firewalls, which often result in slow delivery speeds and is also error-prone, all solutions need to involve automation as much as is practical. Networks as an example are best automated in areas like helping to identify, locate and remove obsolete or unused firewall rules for instance. Any large and complex infrastructure deployment lacking automation requires a lot of personnel to deal with large queues of security incidents to resolve.

It is necessary in hybrid environments to eliminate the traditional slow and passive response common with traditional operational methods of security teams by using automation. A list of automation benefits for a security solution includes:

1. Eliminating misconfiguration by human errors
2. Large scale consistent security orchestration
3. Active and responsive remediation plans
4. Automatic security actions
5. Continuous security testing, verification and auditing

Figure 30 illustrates a model infrastructure orchestration that can be adapted for delivering security as code across infrastructure elements in an enterprise environment using common DevOps tools such as Visual studio Code, it and CI/CD pipelines. Shown below is an example of how to securely orchestrate Kubernates containers on Azure.

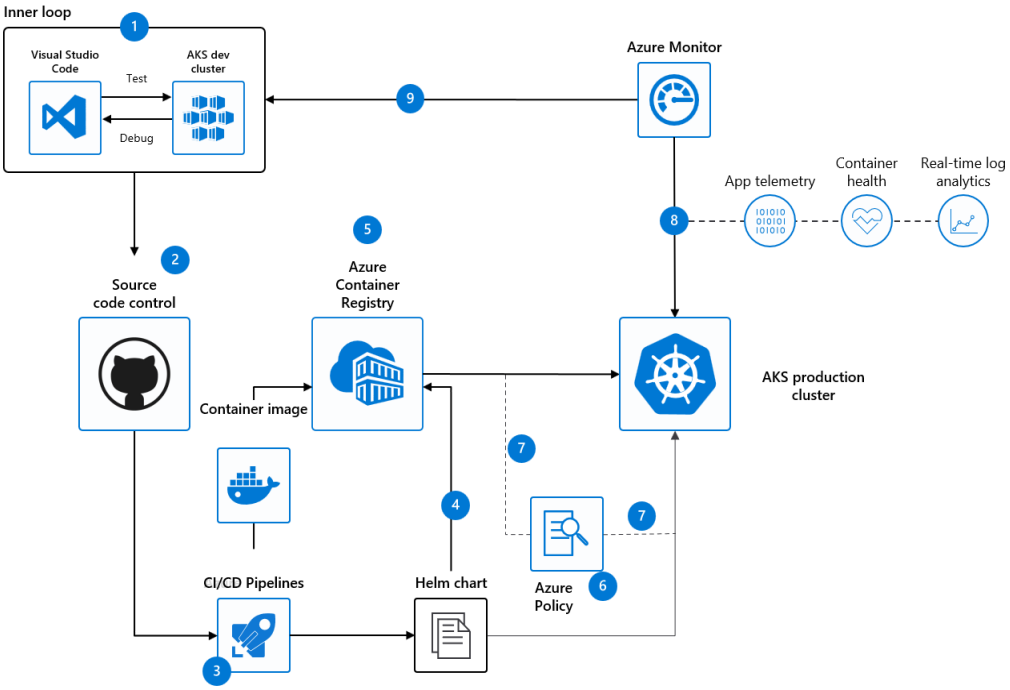

Figure 30: Setting a secure container orchestration [Microsoft Azure]

Automation as a security enabler can easily be seen bridging the pieces of hybrid infrastructure deployment, as it answers to the fulfilling requirements in combination with both technologies and methodologies of managing security with code.

DevSecOps as an automation enabler solve many security challenges of having security embedded in infrastructure deployment process. Coupling Infrastructure delivery pipelines with security appended to deployments can guarantee infrastructures security integrity. Applying automation to a security processes results in being able to identify, validate, and remediate threats. Repetitive tasks and complex tasks are best executed within the context of an automated IT. Automated IT security reduces platform security costs. Automation is pivotal in achieving security at scale [23]. Typically configuration management processes are well automated with configurations neatly encapsulated in the form of code or scripts. Scripts have to be secured and controlled through the version control system such as Git, Azure repos or Bit bucket etcetera. Figure 31 illustrates steps for building a security code ecosystem to embed in a DevOps process.

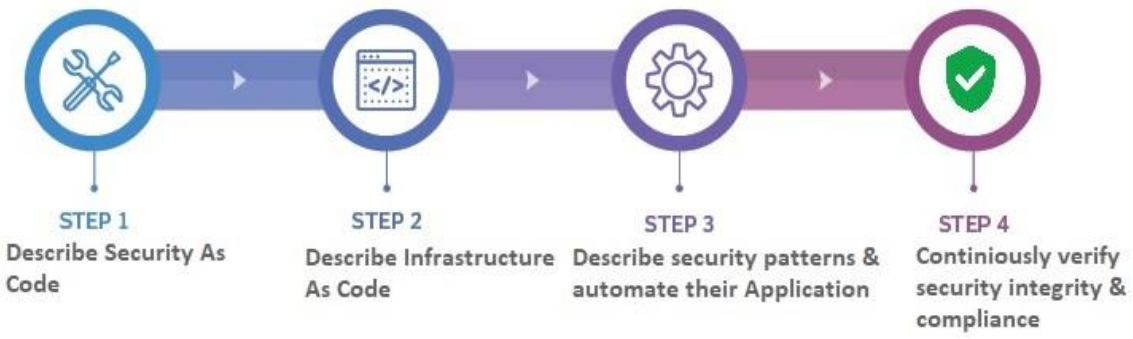

Figure 31: Steps for automating hybrid security with Security as Code

A security map of an environment will enable better visibility to aid in defining a security topology of the entire environment as code. The security definition of an environment generally includes, the setup of servers, network configurations, firewalls, access lists, security policies etcetera. What is required is a code description of where and what is the desired security state affixed to an infrastructure component such as a disk, network interface etcetera. This information is scriptable in code and can then be stored into a version control system where it can be pulled from when required to run as part of an orchestration process for applying security to infrastructure during deployment or remediation.

6.2 Security Automation Tools

Having established that automation is a security enabler in hybrid deployments, selecting the right tools for security automation is vital. Other common tools for secure, automated configuration management and provisioning of infrastructure includes tools such as Chef, Puppet and Docker. Code-driven configuration management tools such as Puppet, Chef or Python make it easy to set up standardized configurations across hundreds of servers (IaaS) using common templates. Figure 32 shows a typical setup.

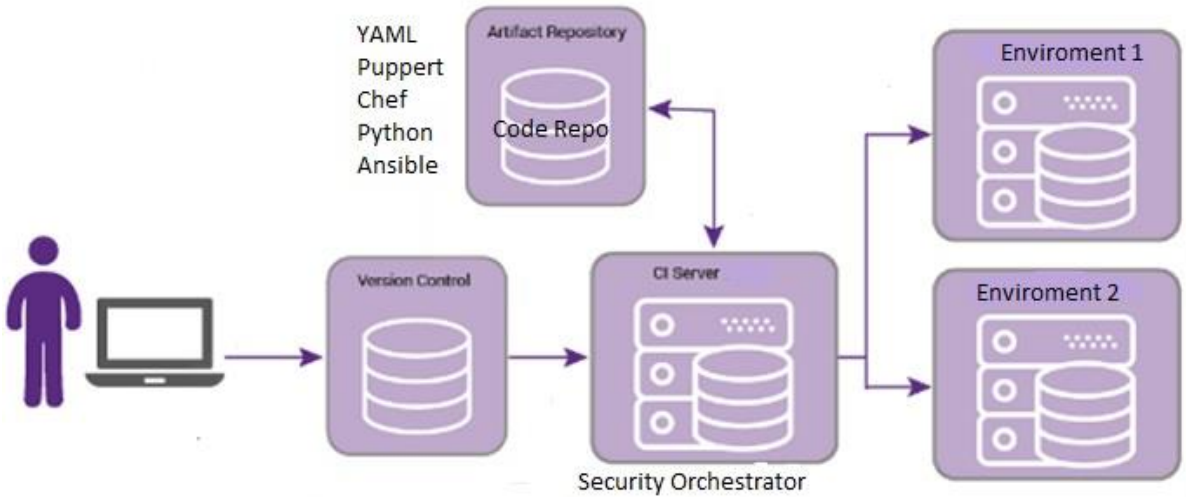

Figure 32: Typical example of Setting up Code-based Security Automation.

The choice of tooling in hybrid infrastructure management is important to get right. Listed below are some tools and Processes required to enable DevOps for security.

- CI/CD tool.
- Secrets manager
- Version control system.
- Security analysis tools.

DevSecOps needs to be set up as an orchestration pipeline. A test environment is created automatically using IaC such as Terraform or Cloud Formation. Chef or Puppet as security configurations tools are then used to describe automation suites to perform security validation of component to verify configurations. This is one way of implementing continuous verification and desired state configuration in a DevSecOps way.

6.3 Security Orchestration, Automation and Response Tools (SOAR)

The taxonomy of hybrid Security architecture is an architectural challenge in large part due to the fact that the security principles in traditional enterprise environments are not implemented using the same technologies used in cloud environments. To address the shared security responsibility model, security incident management needs to be managed by capable tooling such as SIEM and SOAR system to protect our cloud. Figure 33 shows the process flow of a security response mechanism that employs SOAR or SIEM to evaluate logged security incidents with an infrastructure environment and process them for an automated response.

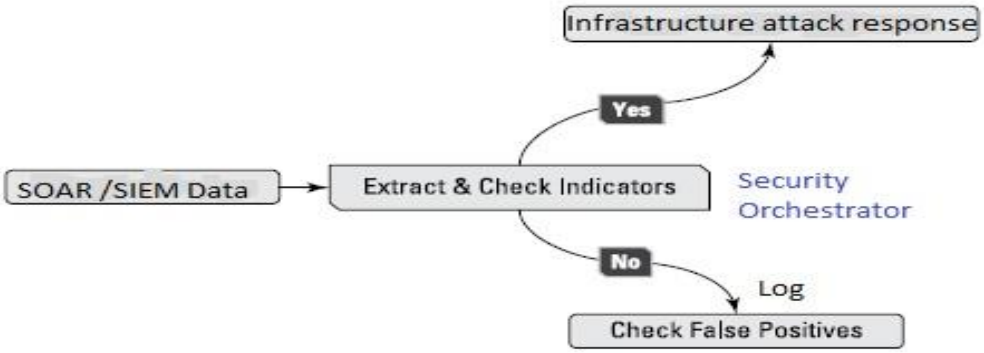

Figure 33: Example Orchestration flow of an automated response system

Figure 33 shows how SOAR or SIEM platforms can be used for extracting security incidents collected by monitoring tools, which can then be feed to an automated security responses engine, knowing the types of security incidents and which infrastructure component is compromised. Machine learning can also be employed for evaluation.

6.3.1 Security Automation with Seem SOAR/SIEM Integration

Considering the scope of security discussion, a cloud adapt risks management strategy provides great impact on the security architecture. Risk management solutions may require developing a continuous benchmarking of security compliance and automatic reporting for frameworks such as CIS, NIST, SOC 2, GDPR, HIPPA, and PCI.

6.4 Enabling Security as Code with DevOps

Orchestrating and managing cloud infrastructure as highlighted in Section 6.1.1 requires automation to achieve security efficiency. Infrastructure as Code DevOps tools such as Jenkins and Azure DevOps shown in Figure 34 as an example of orchestrating security as Code have proved adapt as security automation enablers. A modular approach to infrastructure security configuration is recommended. The aforementioned eliminates lengthy complex code which is hard to manage, update or maintain.

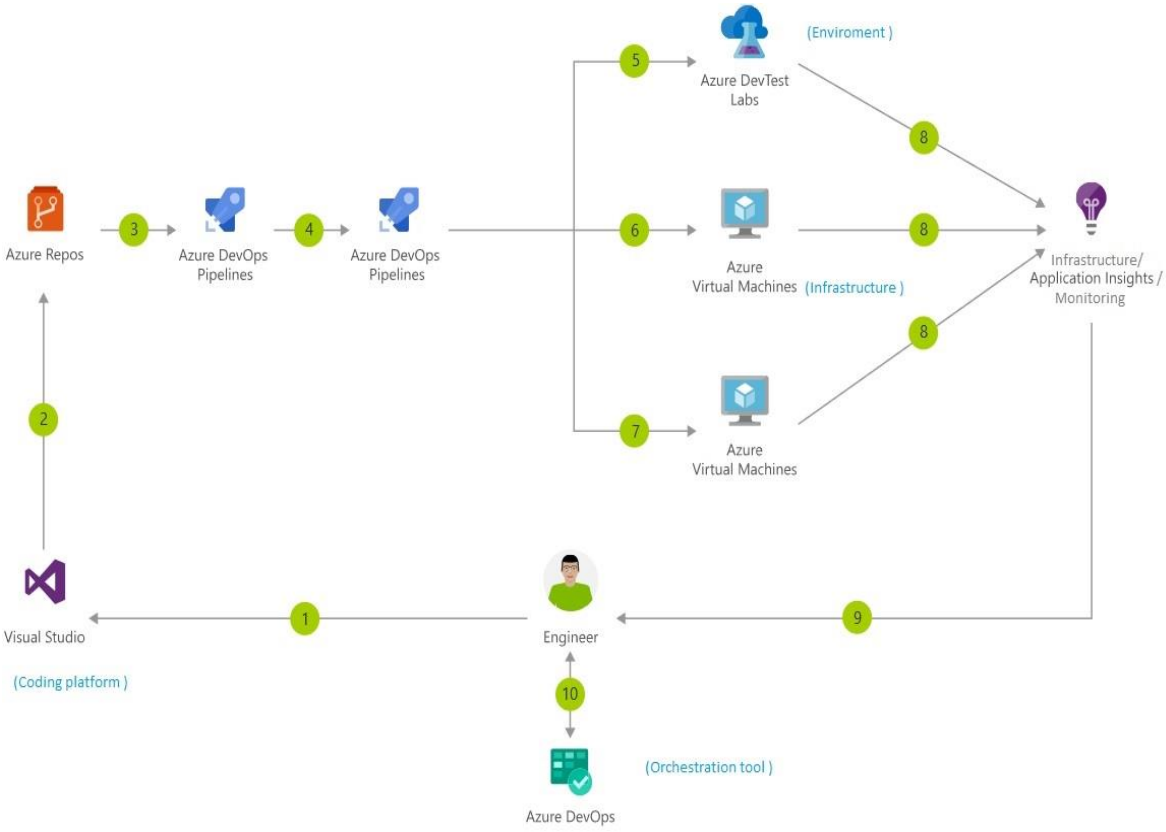

Figure 34: DevOps CI/CD for VM (Infrastructure) provisioning [Microsoft Azure]

The above example demonstrates the concept of Infrastructure as Code delivery process with DevOps. The setup can run security as Code separately or have the security code run at the same time as infrastructure deployment. Since security is a continuous activity, the security pipelines have to run concurrently or embedded into the IaC deployment process. Security as Code therefore also becomes a continuous deliverable process which can be aided by DevOps automation workflows for security compliance.

Listing 4 provides a insight into coding security into the infrastructure deployment code. The approach of achieving challenges of implementing component security depend on the complexity of an IT environment. Code based security definition has to be designed in such a way that it can easily be updated or changed when a functionality has to change configurations or settings. The common cases for example includes building networks, security groups, firewall rules and access settings on virtual network interfaces.

Listing 4: Ansible Automation Code example

```
---
- name: "List pool members"
  hosts: lb
  gather_facts: false
  connection: local

  tasks:
    - name: Query BIG-IP facts
      bigip_device_info:
        provider:
          server: "{{private_ip}}"
          user: "{{ansible_user}}"
          password: "{{ansible_ssh_pass}}"
          server_port: 8443
          validate_certs: false
        gather_subset:
          - ltm-pools
      register: bigip_device_facts

    - name: "View complete output"
      debug: "msg={{bigip_device_facts}}"

    - name: "Show members belonging to pool"
      debug: "msg={{item}}"
      loop: "{{bigip_device_facts.ltm_pools |
json_query(query_string)}}"
      vars:
        query_string: "[?name=='http_pool'].members[*].name[]"
```

Considering a network component such as a network interface card, it can be built as a separate code module mapped to security code that should be applied on the component when provisioned. Post configuration verification can be done through a desired state check that can be described say in a YAML file and administered by for example Ansible configuration management or any scripting language such as Python.

7 Discussion

Public cloud providers in their security implementations have attempted to meet NIST standards for cloud security by developing hybrid solutions based on technologies, advanced identities and modern methods of authentication. Security as a managed service largely the primary driver of hybrid security posturing. What has been discovered is that the integrity of infrastructure security is best orchestrated throughautomation, as automation eliminates errors and speed up configurations and security response time. Hybrid security implementations have a high requirement for architectural approaches for securing infrastructure across the private-public IT boundaries. The trending approaches in computing models has greatly shifted as predicted by research firms such as Gartner and Forrester. The aforementioned computing shift has given rise to new methods of implemented security across IT boundaries. There are many areas of common approaches to security integration and management across all cloud providers. The common approaches include removing complexity and employing automation. Hybrid IT has become synonymous with automated self-healing infrastructure deployment and management by code (IaC). Deployment orchestration methodologies are increasingly evolving into process based standards such as DevOps with embedded security commonly referred to as DevSecOps.

7.1 Analysis

In the final analysis, it can be argued that it is not enough to build security around any one pillar or single approach. It is equally difficult to balance optimized infrastructure access with the highest level of security. They are security approaches that are otherwise stronger and better suited for protecting infrastructure than others. It is therefore recommended that the integration of different security technologies should be actualized by way of Security as Code. Infrastructure as Code deployment and automated security management have been proven to reduce security misconfigurations and eliminate security gaps. Security architects need to adopt building infrastructure access based on new forms of Authentication, Authorization and Access (AAA) such as blockchain, compound identities and service principals. A zero trust network security approach coupled with SDN improves security and visibility in hybrid environments.

7.2 Potential Research

Future research in novel hybrid identities promises a security leap. It can be proposed as a security enabler, that in multi-tenant environments and infrastructure such as storage and containers, immutable security be adopted and implemented. Security threats will continue to evolve and so is the risk of data exfiltration which will become more sophisticated in the future. The aforementioned will require security development around new ideas, such as identity and access technologies. Some great areas of security research concepts will include the development of abstract real-time object-oriented security entities. The benefits of compound real-time identities will be the ability to make a session token as just an attribute in a complex security object. This will move security to the next level as it will not be tied to simple credentials. Artificial intelligence (AI) and Machine learning (ML) as part of the security toolbox is recommended. It can be theorized that compound identities will become the redefined credentials of the future. We can postulate that with changes in security technologies, rethinking legacy IT security architecture when building modern hybrid infrastructure needs a code centric security approach. The real test will be whether, a new identity and access paradigm shift will finally render identity theft and data breaches impossible. Security approaches and implementations, like those discussed in this thesis represent an automation centric security approach. There are other approaches however such as those driven by applications or data centric emphasis which have not been covered in this thesis. In summary, infrastructure should be described by code and so should be the security management of hybrid IT.

7.3 Conclusion

Cloud security hyper-resilience can in part be helped if and when identity becomes the new perimeter. For many years the concept of identities as a security perimeter was was not even possible as access was solely by traditional means of user credential combination of user name and password. Credential theft and leakage common and rife in IT, that social engineering and brute force attacks are very common threats.

Whatever the business case for deploying a hybrid cloud, the decision to consume public IT resources through a hybrid integration has no single solution. There are only recommendations and best practices to follow. These recommended best practices are what dictates architecting decisions and solutions in respect of security technologies. The end result is that implementing hybrid security solutions without compromising

security of private IT installations held on-premises by bridging them with a public cloud required putting to work much of what been described in this research. There is a consensus on what security strategy is best to secure hybrid infrastructure and data. Central to this consensus, is the importance of identity and access management in hybrid IT. Useful recommendations are here listed in the appendix.

References

1 Bond J, The enterprise cloud: best practices for transforming legacy IT, First edition. O'Reilly. 2015

2 Charles J. Brooks, Christopher Grow, Philip Craig, Donald Short, Cybersecurity Essentials, John Wiley & Sons, 2018

3 Red Hat's approach to hybrid cloud security https://www.redhat.com/en/topics/security/hybrid-cloud-security-approach. accessed 04.09.2020

4 Defending the New Perimeter, Pete Zerger and Wes Kroesbergen, Modern Security from Microsoft

5 Rose, S. , Borchert, O. , Mitchell, S. and Connelly, S. (2020), Zero Trust Architecture, Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, [online], https://doi.org/10.6028/NIST.SP.800-207, https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=930420 (Accessed June 3, 2021)

6 Google. (n.d.). Hybrid and multi-cloud network topologies; Cloud Architecture Center. Google. https://cloud.google.com/architecture/hybrid-and-multi-cloud-network-topologies. accessed 04.09.2021

7 Schuba, C.L. & Spafford, Eugene. (1998). A reference model for firewall technology. 133 - 145. 10.1109/CSAC.1997.646183.

8 Hybrid Cloud Security, VMware, https://www.vmware.com/topics/glossary/content/hybrid-cloud-security, accessed 04.09.2021

9 Leskinen, Jesse. "Evaluation criteria for future identity management." 2012 IEEE 11th international conference on trust, security and privacy in computing and communications. IEEE, 2012.

10 Architecting the Cloud : Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS) by Michael J. Kavis, John Wiley & Sons, Incorporated, 2014-01-28

11 AWS Well-Architected Framework. 2020 Amazon Web Services

12 Ortega, R. (2006). DEFENDING THE CORPORATE CROWN JEWELS FROM THE DANGERS THAT LURK WITHIN: EFFECTIVE INTERNAL NETWORK SECURITY FOCUSES ON BEHAVIOR. EDPACS, 34(1), 1-10. https://doi.org/10.1201/1079.07366981/46107.34.1.20060701/93700.1

13 Ohlhorst, F. 2011. Unified Security Management Is A Must Have When Securing Private And Hybrid Clouds. Network Computing - Online

14 Klaffenbach, F. & . 2019. Multi-Cloud for Architects: Grow Your IT Business by Means of a Multi-Cloud Strategy

15 Guidelines on Firewalls and Firewall Policy, Recommendations of the National Institute of Standards and Technology .pdf. published September 2009

16 Identifying Cloud Security Threats to Strengthen Cloud Computing Adoption Framework, The 2nd International Workshop on Internet of Thing: Networking Applications and Technologies (IoTNAT' 2016)

17 Cloud Security Alliance (2011) Security guidance for critical areas of focus in Cloud Computing V3.0. Available https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf. (n.d.).

18 Li W, Ping L, Trust model to enhance Security and interoperability of Cloud environment, Proceedings of the 1st International conference on Cloud Computing. Springer Berlin Heidelberg, Beijing, China, 2009, pp 69–79.

19 Dawoud W, Takouna I, Meinel C, Infrastructure as a service security: Challenges and solutions. The 7th International Conference on Informatics and Systems (INFOS), Potsdam, Germany. IEEE Computer Society, Washington, DC, USA, 2010, pp 1–8.

20 I. Ahmad, S. Namal, M. Ylianttila and A. Gurtov, "Security in Software Defined Networks: A Survey," in IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2317-2346, Fourthquarter 2015, doi: 10.1109/COMST.2015.2474118.

21 Leskinen J: Evaluation criteria for future identity management. In Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on. IEEE, Piscataway, New Jersey, United States; 2012:801–806.

22 Assante, M., Tobey, D. (2011, February 4). Enhancing the Cybersecurity Workforce. Retrieved from http://ieeexplore.ieee.org/document/5708280/

23 Cabuk, S., Dalton, C., Eriksson, K., Kuhlmann, D., Ramasamy, H., Ramunno, G., . . . Stble, C. 2010. Towards automated security policy enforcement in multi-tenant virtual data centers. Journal of Computer Security, 18(1), pp. 89-121. doi:10.3233/JCS-2010-0376

List of Important Security Principles in Hybrid Environments

1. Implement advanced Identities for hybrid environments.
2. Authenticate users and processes
3. Authorize after you authenticate
4. Avoid security by obscurity
5. Check the return value of functions
6. Clearly delineate the physical and logical security boundaries
7. Compartmentalize
8. Requires a Comprehensive and Integrated Approach
9. Computer Security Responsibilities and Accountability Should Be Made Explicit
10. Computer Security should be periodically reassessed
11. Data in transit protection
12. Declare data objects at the smallest possible level of scope
13. Defense in depth
14. Design and implement audit mechanisms
15. Design and operate an IT system to limit damage and to be resilient in response.
16. Design for secure updates
17. Design for security properties changing over time
18. Design reviews
19. Design security to allow for regular adoption of new technology
20. Develop and exercise contingency or disaster recovery procedures to ensure appropriate availability
21. Do not implement unnecessary security mechanisms.
22. Don’t trust infrastructure
23. Don’t trust services (from others)
24. Earn or give, but never assume or trust

Listing 3: Terraform Infrastructure Deployment Code example

```
# Providers
provider "azurerm" {
  version = "~> 2.24"

  subscription_id = var.subscription_id
  tenant_id       = var.tenant_id
  client_id       = var.client_id
  client_secret   = var.client_secret

  features {}
}

# resource group
resource "azurerm_resource_group" "thesis-rg" {
  name     = "thesis-rg"
  location = "West Europe"
}

# network
resource "azurerm_virtual_network" "thesis-net" {
  name                = "thesis-net"
  location                      =  azurerm_resource_group.thesis-
rg.location
  resource_group_name = azurerm_resource_group.thesis-rg.name
  address_space       = ["10.10.0.0/16"]
}

# subnet
resource "azurerm_subnet" "thesis-sub" {
  name                 = "thesis-sub"
  resource_group_name = azurerm_resource_group.thesis-rg.name
  virtual_network_name = azurerm_virtual_network.thesis-net.name
  address_prefixes     = ["10.10.10.0/24"]
}

# nics
resource "azurerm_network_interface" "thesis-nic" {
  name                = "thesis-nic"
  location                      =  azurerm_resource_group.thesis-
rg.location
  resource_group_name  = azurerm_resource_group.thesis-rg.name
  enable_ip_forwarding = true

  ip_configuration {
    name                          = "thesis-nic-ip-config"
    subnet_id                     = azurerm_subnet.thesis-sub.id
    private_ip_address_allocation = "Static"
    private_ip_address            = "10.10.10.1"
    public_ip_address_id             = azurerm_public_ip.thesis-
ip.id
  }
```

```
}

# public ips
resource "azurerm_public_ip" "thesis-ip" {
  name                = "thesis-ip"
  location                      =  azurerm_resource_group.thesis-
rg.location
  resource_group_name = azurerm_resource_group.thesis-rg.name
  allocation_method   = "Static"
}

# network security group
resource "azurerm_network_security_group" "thesis-nsg" {
  name                = "thesis-nsg"
  location                      =  azurerm_resource_group.thesis-
rg.location
  resource_group_name = azurerm_resource_group.thesis-rg.name

  security_rule {
    name                       = "SSH"
    priority                   = 100
    direction                  = "Inbound"
    access                     = "Allow"
    protocol                   = "Tcp"
    source_port_range          = "*"
    destination_port_range     = "22"
    source_address_prefix      = "*"
    destination_address_prefix = "10.10.10.1"
  }
}

resource     "azurerm_subnet_network_security_group_association"
"thesis-sub-nsg-assoc" {
  subnet_id                 = azurerm_subnet.thesis-sub.id
  network_security_group_id                 =                az-
urerm_network_security_group.thesis-nsg.id
}

# vms
resource "azurerm_virtual_machine" "thesis" {
  name                          = "thesis"
  location                       = azurerm_resource_group.thesis-
rg.location
  resource_group_name           = azurerm_resource_group.thesis-
rg.name
  network_interface_ids                                    =    [az-
urerm_network_interface.thesis-nic.id]
  vm_size                       = "Standard_DS1_v2"
  delete_os_disk_on_termination = true

  storage_image_reference {
    publisher = "Canonical"
    offer     = "UbuntuServer"
```

```
    sku       = "19_10-daily-gen2"
    version   = "latest"
  }

  storage_os_disk {
    name              = "thesis-osdisk"
    caching           = "ReadWrite"
    create_option     = "FromImage"
    managed_disk_type = "Standard_LRS"
  }

  os_profile {
    computer_name  = "thesis"
    admin_username = var.admin_username
  }

  os_profile_linux_config {
    disable_password_authentication = true
    ssh_keys {
      key_data = file("~/.ssh/id_rsa.pub")
      path                                                    =
"/home/${var.admin_username}/.ssh/authorized_keys"
    }
  }
}
```

List of Security recommendation for architecting hybrid security

1. Security tooling is everything required to begin planning Hybrid Cloud Architecture. Cloud providers offer self-assessment tools which can be a starting point for planning.
2. Whereas there is no single way of implementing IT or security solutions, often getting the approach correct and adhering to cyber principles and security practices and frameworks will set up a bases for a solution.
3. Using a firewall manager for centralized network security policy and route management for globally distributed, software-defined perimeters by means of software defined networks is a great approach for the next generation security solutions.
4. Micro-segmentation is advised to help address the problem of privilege escalation which is considered to be the most serious security threat for running containers at scale in enterprise production.
5. Many security professionals recommend running containers within a hypervisor-driven VM environment for better isolation through micro-segmentation as a means to prevent potential privilege escalation.
6. Use proven technologies and tested integrations, third party system integrations always adds complexity and carful assessment has to be made why native cloud security solution is not adequate for a deployment scenario.
7. Use modern methods of authentication with managed identities to enhance hybrid security.
8. Build the hybrid environment as Infrastructure as code and automate security orchestration and management.
9. A modular environment is easier to secure and effectively test, monitor, segment and remediate. Therefore, it stands as the best approach to building secure infrastructure.
10. As no infrastructure secure is ever 100% secure, Connections, and data itself need to be secured by encryption. This also includes applications that are deployed on the infrastructure.