



# Virtualisation/Cloud Security

---

Barbara Krasovec, EGI CSIRT, JSI

Split, June 2022

# Virtualisation

---

Virtualisation architecture is the abstraction of physical resources, hypervisor sits on top of physical hardware and abstracts physical resources.

# Virtualisation security

---

Security procedures and controls for all components of virtualisation infrastructure.

## Virtualisation and cloud

---

- **virtualisation is a technology**: it allows creating multiple environments from a single, physical hardware system
- **cloud is an environment**: it can include bare-metal, virtualisation, or container software

# Virtualization and cloud (2)

---

	Virtualization	Cloud
<b>Definition</b>	Technology	Methodology
<b>Purpose</b>	Create multiple simulated environments from 1 physical hardware system	Pool and automate virtual resources for on-demand use
<b>Use</b>	Deliver packaged resources to specific users for a specific purpose	Deliver variable resources to groups of users for a variety of purposes
<b>Configuration</b>	Image-based	Template-based
<b>Lifespan</b>	Years (long-term)	Hours to months (short-term)
<b>Cost</b>	High capital expenditures (CAPEX), low operating expenses (OPEX)	Private cloud: High CAPEX, low OPEX Public cloud: Low CAPEX, high OPEX
<b>Scalability</b>	Scale up	Scale out
<b>Workload</b>	Stateful	Stateless
<b>Tenancy</b>	Single tenant	Multiple tenants

Source: <https://www.redhat.com/en/topics/cloud-computing/cloud-vs-virtualization>

# Why virtualising?

---

- efficient usage of resources
- lower operating costs (compared to using physical machine for each service)
- flexibility

# Why does cloud security matter?

---

- hypervisors are prime targets of attacks (single point of failure)
- if hypervisor host is vulnerable, everything else on it is vulnerable
- VMs can interfere with each other
- resources and services are difficult to track
- lack of knowledge of technical staff
- data is sparsed on multiple servers and locations
- all security risks present in traditional infrastructure are also present here

# Virtualisation security essentials

---

- don't use default credentials
- don't mix production and development VMs on the same hypervisor, use different network or at least different security group for production and development
- use different credentials for production and development VMs
- monitor all VMs (production, testing, development)
- shut down VMs that you don't need
- always update offline VMs before putting them back online
- maintain inventory of VMs
- check for open ports, default passwords, unpatched software (nmap, Metasploit, OpenVAS, Nessus) - check also <https://github.com/dev-sec/puppet-os-hardening>



# Hypervisor

---

Hypervisor is the main component in the virtualisation.

- hypervisor controls access between virtual guests and host hardware

# Hypervisor security

---

Same security recommendations as for any other host:

- keep your software updated
- remove/mask services that you don't need
- if possible use the same hardware for all hosts (easier to follow vulnerabilities from just one vendor, secure boot)
- apply HW firmware updates before OS installation
- restrict access to hypervisors and monitor it (do you really need a public IP?)
- use restrictive SSH access settings
- use SELinux (put a hypervisor in another security context)
- audit and use firewall

## Hypervisor security (2)

---

- monitor integrity of executables - FIM (e.g. AIDE)
- encrypt communication between core services
- apply rate limiting on incoming connections
- access via management network
- restrict access (SSH key to login, VPN access, packet filtering)
- disable direct hardware access from guests (PCI, USB etc.)
- guests should not have access to host and other guests

## Hypervisor security (3)

---

Hypervisors in the public cloud context even bigger targets for attacks

# Cloud security

---

# Cloud services

---

Consider the benefits of running services in the cloud.

- What are your risks?
- What are your responsibilities?
- Which domains are under your control and which in the hands of the cloud provider?
- Where will you store your data and how will you transfer it, use it?
- Are there any regulations about storing the data in the cloud?

# Cloud services

---


There are 3 main types of as-a-Service solutions:


- IaaS - infrastructure as a service (VMs)
- PaaS - platform as a service (DB,k8s)
- SaaS - software as a service (applications)

# Cloud models

---

On-site	IaaS	PaaS	SaaS
Applications	Applications	Applications	Applications
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
O/S	O/S	O/S	O/S
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking

 You manage

 Service provider manages



## Cloud security challenges

---

- for customer: no longer access to the hypervisor or hardware (physical, host security), cannot control which customers host on the same host and how well they protect their VMs
- for cloud provider: complex network designs and no control over the state of VMs

# Common threats in the cloud

---

- cyber attacks: DoS, spoofing, man-in-the-middle
- escalation of privileges, unauthorized access
- hijacking accounts
- misconfigurations
- internal/external threats
- malware
- data breaches
- insecure interfaces/APIs
- external data sharing and data transfers
- insufficient technical skills
- VM escape
- leaked credentials (committed to git)

# Unauthorised access

---

To gain unauthorized access, attackers need to:

- gain access to VM
- gain access to the host from the VM

To prevent this:

- customers need to keep VMs updated
- providers need to implement virtual networks, keep hypervisors updated and prepare a good cloud design

# Cloud design

---

- which services need to be access from the Internet?
- OpenStack relies on APIs that are accessible from the Internet
  - how to protect APIs?
- how to protect hypervisor?
- which authentication and authorization mechanisms to implement?
- how to provide secure communication between the cloud components?

# Levels of cloud security

---

## CLOUD SECURITY

1

### PHYSICAL SECURITY

Cloud provider is responsible for the physical security of the servers and devices.



2

### INFRASTRUCTURE SECURITY

Cloud provider is responsible for hardware, network and hypervisors.



3

### PLATFORM SECURITY

Shared responsibility between customer and cloud provider



4

### APPLICATION SECURITY

Responsibility of the customer



5

### DATA SECURITY

Responsibility of the customer.

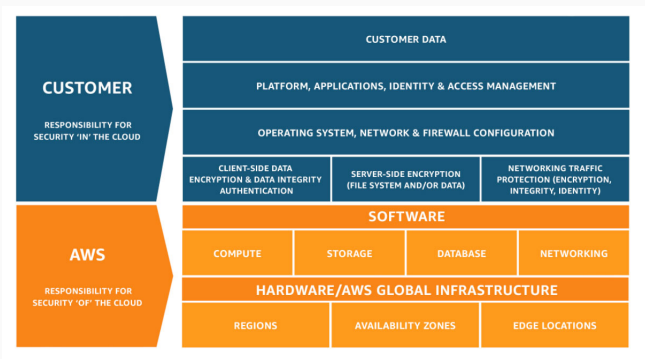


## Difference public - private cloud

---

# Shared responsibility

---



Source: <https://aws.amazon.com/compliance/shared-responsibility-model/>

# Public vs private cloud security

---

- **Private cloud:**
  - security is a responsibility of the organisation
  - number of VMs is pretty stable
  - scalability is limited
  - bandwidth is limited
  - data storage and access under control of the organisation
  - potential of providing perfectly safe environment (behind a firewall)
- **Public cloud:**
  - shared responsibility between customer and cloud provider
  - seemingly infinite resources
  - main target for security attacks (security is big investment)
  - no control over data for customer
  - customer needs to trust cloud provider



# Private cloud security considerations

---

- understand you security weaknesses
- monitor your network
- BIOS hardening and updates
- restrict access to IPMI/ILO interfaces, keep them updated
- security groups are not the same as ACLs, set allowed traffic per VM and name the security groups in such way that the rule will already explain what it does (e.g. SMTP\_IMAP\_ACCESS\_TO\_NETWORK\_ABC)
- all inbound connections should use TLS, all service to service connections should use TLS
- access to management nodes limited (from certain IPs, by VPN, bastion host ..)
- outbound filtering with egress
- use rate limiting on incoming connections

# Private cloud security considerations

## (2)

---

- use management network for core cloud services, data network for storage nodes, APIs network for APIs, external network for VMs and internal network for VM to VM communication within the same cloud
- encrypt your data
- use role-based access control (define API policy per service)
- always keep an off-cloud backup
- perform regular updates
- automate deployment and configuration of services

# Infrastructure security in public cloud

---

- Regions (physical location of the clusters)
- Availability zones (isolated areas within each region - independent power, cooling, and physical security)
- protecting network: zero trust approach (network and account boundaries)
- network connectivity includes thousands of VPCs, accounts, and on-premises networks (AWS Transit Gateway - acts as a hub that controls how traffic is routed among all the connected networks)
- define system security configuration and maintenance (hardening, minimization and patching)
- operating system authentication and authorizations (for example, users, keys, and access levels)

# Infrastructure security in public cloud (2)

---

- define web application firewalls and/or API gateways
- automate network protection: self-defending network (threat intelligence, anomaly and intrusion detection (WAF))
- limit access (who can access what, from where, for how long)
- perform vulnerability management
- automate creating resources (eg. AWS CloudFormation secure templates) and configuration
- provide OS hardening

# Public cloud security tools

---

- each public provider has its own set of security tools
- hardening the cloud resources comes with a price
- being target of multiple attacks, public cloud are generally secure
- monitoring of events is already integrated in cloud provider's tools
- cloud security should follow NIST's five pillars of a cybersecurity framework: Identify, Protect, Detect, Respond, and Recover

# Example: AWS Cloud Security

---

- access policy: avoid using root, use MFA, disable credentials unused for 90 days or less, ensure access keys rotation every 90 days, IAM password policy (same as for any OS)
- automate building resources and their configuration
- use multilayered network
- perform regular backups
- AWS has a lot of security tools:
  - AWS config: asses, audit of configuration of AWS resources
  - AWS CloudTrail: checks changes, records AWS API calls
  - AWS Config: configuration management of supported AWS resources in your account
  - AWS CloudWatch: applications monitoring (visualisation of metrics, logs, events)
  - AWS Guard Duty: threat detection system
  - AWS Shield: dDos protection

# Cloud network security

---

- new infrastructure can be instantly added by any person or system with no expert skills
- ease of deployment and high rate of change make it very difficult to maintain overall control over cloud environment (autoscaling, serverless computers)
- customer shares responsibility with the provider for securing network
- baseline: educating dev-ops and users, establishing who is responsible for which aspect of security, the use of CIS benchmarks, incident response plan
- use tools to monitor and detect vulnerabilities and misconfigurations in cloud network
- use SIEM or threat detection solution

## How about in private cloud?

---

Here the network design is in the hands of the organisation.  
It needs to be carefully planned.



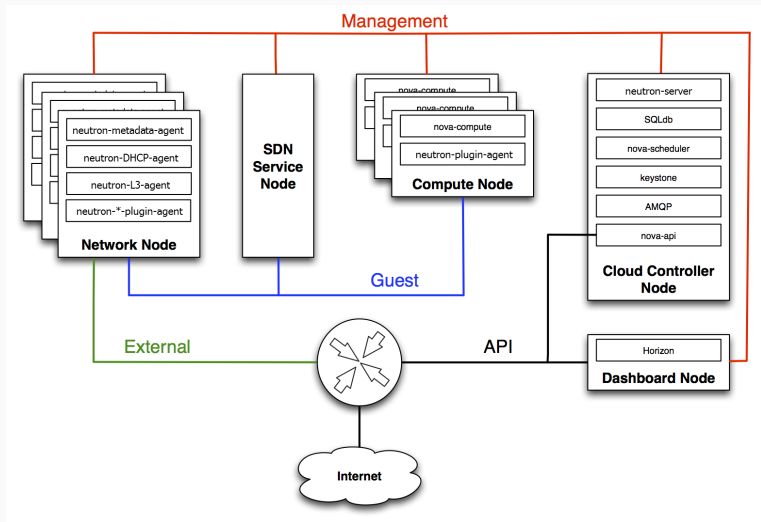
# OpenStack network segmentation

---

OpenStack uses SDN, which complicates the design of physical and virtual networks.

- There are typically 4 types of network in OpenStack:
  - **API network:** used to access APIs, accessible by anyone from the internet
  - **Management network:** used for communication between the OpenStack components, traffic is typically not routed in or out of this network. (databases)
  - **Guest/tenant network:** Used for VM data communication within the cloud deployment.
  - **External/public network:** reachable from the Internet.

# OpenStack network design example



# Network security

---

- Firewall: whitelisting between zones, perimeter control, separating your systems from the rest of the world and internal segmentation
- ACLs: rules for each network
- Security groups: Similar to network ACLs - security group rules implemented as a service, they apply at a per-OS instead of per-network

# Node provisioning

---

- use PXE boot for installations
- use separate network for PXE
- use configuration management
- verify boot process (secure boot)
- use multitenant network

# Harden OpenStack Configuration

---

- perform regular updates (also servers in private network)
- automatic deployment of services
- all VMs should be able to migrate to another hypervisor with no downtime
- don't use local disks for VMs, use distributed storage (as Ceph) - RBDs are easily connected to another host
- configs should be read-only

## Secure communication

---

- use TLS for communication between components
- put APIs behind TLS proxy (Nginx, Apache, Pound, Stud)
- secure http endpoints with strict transport security (HSTS), set max-age to one day or one week, then extend

# Authentication and Authorization

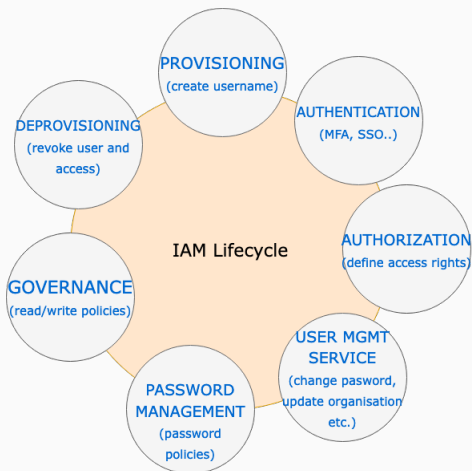
---

- authentication (identity) vs authorization (access)
- traditional IT: getting identity means obtaining email, VPN, access to services
- cloud: deleting identity means not being able to login, how about access?
- services provide long-lived authentication tokens that exist even when identity is gone

# IAM lifecycle

---

IAM = identity and access management





# Private cloud APIs

---

APIs are the point of contact with the outside world, so firewall is not enough

- usernames, passwords and tokens should never appear in the URL, use proxy caching and logging, data encryption
- isolate API endpoints on separate hosts
- use SELinux
- use host-based firewall rules
- use network ACLs and IDS
- use two factor authentication where possible
- use reverse proxy for REST API endpoints
- use WAF (Web application firewall to protect your APIs against common web exploits

OpenStack has private and public API-s, by default they are all public, but they don't have to be. Specify allowed API operations

## Public cloud API

---

- in AWS all are public by default
- Service Control Policies: where you make limitations for APIs, whitelists, limits to regions..

# Secure VMs

---

- follow OS hardening guidelines (STIG and CIS controls)
- harden automatic installation and configuration (e.g. Ansible)
- remove services and packages
- provide security monitoring (e.g. Prometheus)
- track versions of OS, software, users with access, images
- use asset management: keep track of what is (still) needed, if not delete
- vulnerability management

# VM attacks

---

- hypervisor breakout (VM escape) - exploit of software vulnerability
- side-channel attacks (eg. Spectre, Meltdown) - unintended side effect of running code on physical system - this is the cloud provider's responsibility
- misconfigurations (e.g. network firewall rules exclude localhost)

# VM images

---

- Don't store credentials, certificates or any sensitive data in the image
- Sign images
- Don't use third party images
- Scan images for vulnerabilities and keep them updated
- Track images (asset management)

# Cloud compute nodes security

---

- don't store credentials in plain text files (such as OS\_USERNAME, OS\_PASSWORD in OpenStack - better to use OpenStack CLI on a separate host)
- use tokens and limit their validity as much as possible (default in OpenStack is 1 hour, in older releases it was 24 hours)
- disable bash history
- store all logs remotely
- disable PCI passthrough
- disable memory optimization as it uses memory pages deduplication
- use TLS for Spice/VNC sessions
- Don't keep VM logs that could contain any sensitive data from the customer

# Cloud Data Security

---

Cloud has multiple data stores: object storage, block storage and file storage

- Object store is like a valet parking: you give a car to a valet, he parks and gives you a ticket, you don't care where the car is parked (files as objects, application manages them, not caring about where they stay and how big they are)
- Block storage as traditional hard disks (FC, iSCSI)
- File storage presents itself as filesystem (NFS, CephFS)

One is a fact: data will move between different physical nodes

# Cloud data foundational security strategies

---

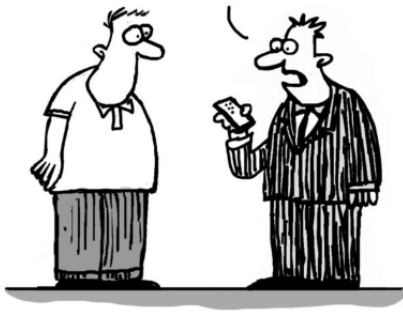
- encryption: protect data at rest, in transit, and in use
- key management includes creating, distributing, storing, making recovery and revoking keys, where encryption keys are stored can affect the overall risk of the data
- obscuring data in the cloud by masking, obfuscation, anonymization, and tokenization
- monitor activities, detect unusual events
- to better collect, manage, analyze, and display log data, use SIEM



# Cloud data

---

NAH, I'M NOT  
WORRIED ABOUT CLOUD  
SECURITY. MY STORED  
DATA IS SO DISORGANIZED  
THEY'D NEVER BE ABLE TO  
FIND ANYTHING!



# How to prevent common attacks?

---

- **Spoofing**: use SSH keys for authentication, TLS for communication, strong password policy, link Keystone with LDAP directory
- **Tampering**: use digital signatures for data integrity (Glance supports image signing), mandatory access control (MAC) and role based access control (RBAC) to protect services
- **Reputation**: central logging and auditing in place, SIEM, monitor networks of anomalies (IDS/IPS)
- **Data disclosure**: use encryption, MAC/RBAC
- **DoS**: redundant services (HA), use quotas per domain/project/user, isolate services from direct access, use proxy to access services from DMZ, good network design
- **Escalation of privileges**: MFA, restrict API, monitor

**Questions?**

# References

---

- Aditya K. Sood: Empirical Cloud security, Mercury Learning
- Chris Dotson: Practical Cloud security, O'Reilly Media
- Fabio Alessandro Locati: Openstack cloud security, Packt Publishing
- Ben Malisow: CCSP Certified Cloud Security Professional Official Study Guide, Sybex
- Silvano Gai: Building a future-proof Cloud Infrastructure
- Chris Binnie, Rory McCune: Cloud Native Security, Wiley Publishing
- Ben Silverman and Michael Solberg: OpenStack for Architects, Packt Publishing
- Donald A. Tevault : Mastering Linux Security and Hardening, Packt Publishing
- James Turnbull: Hardening Linux, APress

## References (2)

---

- RedHat Openstack Hardening guide:  
[https://access.redhat.com/documentation/en-us/red\\_hat\\_openstack\\_platform/16.2/html/security\\_and\\_hardening\\_guide/index](https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/16.2/html/security_and_hardening_guide/index)
- OpenStack Security Guide:  
<https://docs.openstack.org/security-guide>
- Cloud Security Alliance:  
<https://cloudsecurityalliance.org>