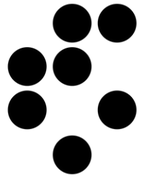


OCTOBER 2023

VIRTUALISATION AND CLOUD SECURITY

Barbara Krašovec

TABLE
of contents

Virtualisation

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

Why virtualising

- efficient usage of resources,
- multiple OSs on the same system,
- isolation of services,
- lower operating costs (compared to using physical machine for each service),
- flexibility,
- fast (re)deployment,

Virtualisation security

- first establish a context, where you identify the risks and assess them,
- consider that it is not appropriate to host virtual instances with widely differing risks and impacts of compromise on the same virtualisation platform/host,
- harden both hypervisors and VMs,
- provide network segmentation and isolation of services,
- define security procedures and security controls for all components of virtualisation infrastructure,
- schedule backups,
- make sure your hosts are running the latest firmware and that all software is updated regularly.

Virtualisation security essentials

- don’t use default credentials,
- don’t mix production and development VMs on the same hypervisor, use different networks or at least different security groups for production and development,
- define different risk profiles and group services on the same host, based on the assessment,
- 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

A hypervisor a software that you can use to run multiple virtual machines on a single physical machine

Hypervisor

- Hypervisor is the main component in virtualisation,
- the hypervisor controls access between virtual guests and host hardware,
- once a hypervisor is compromised, the attacker owns everything,
- access control and monitoring of virtual administrators is critical,
- there is no way to completely isolate one OS from another (no physical separation as in physical networks) - direct memory access, resource sharing

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, use 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 configuration,
- use SElinux (put a hypervisor in another security context),
- audit and use firewall.

Hypervisor security

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

Hypervisors in the public cloud context are even bigger targets for attacks

VM security

- follow OS hardening guidelines (STIG and CIS controls),
- harden automatic installation and configuration (e.g. Ansible),
- remove services and packages that are not needed,
- 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,
- implement vulnerability management.

VM attacks

- hypervisor breakout (VM escape) - exploit of software vulnerability, rare (but CVE-2008-0923, CVE-2022-20779),
- VM hyper jumping by exploiting the host/hypervisor,
- DoS,
- side-channel attacks (eg. Spectre, Meltdown) - an unintended side effect of running code on the physical system breaks encryption - this is the cloud provider's responsibility,
- misconfigurations (e.g. network firewall rules exclude localhost),
- code execution,
- non-updated hypervisors with outdated packages,
- hyperjacking - by exploiting a vulnerability in the hypervisor the attack targets VMs

VM images

- Don’t store credentials, certificates or any sensitive data in the image,
- harden the software in the image,
- sign images,
- don’t use third-party images (use trusted sources),
- scan images for vulnerabilities,
- keep them updated
- track images (asset management).

Virtualisation and cloud

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

Virtualisation and cloud

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

Running services in the cloud

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, and use it?
- Are there any regulations about storing the data in the cloud?

Cloud implementation models

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

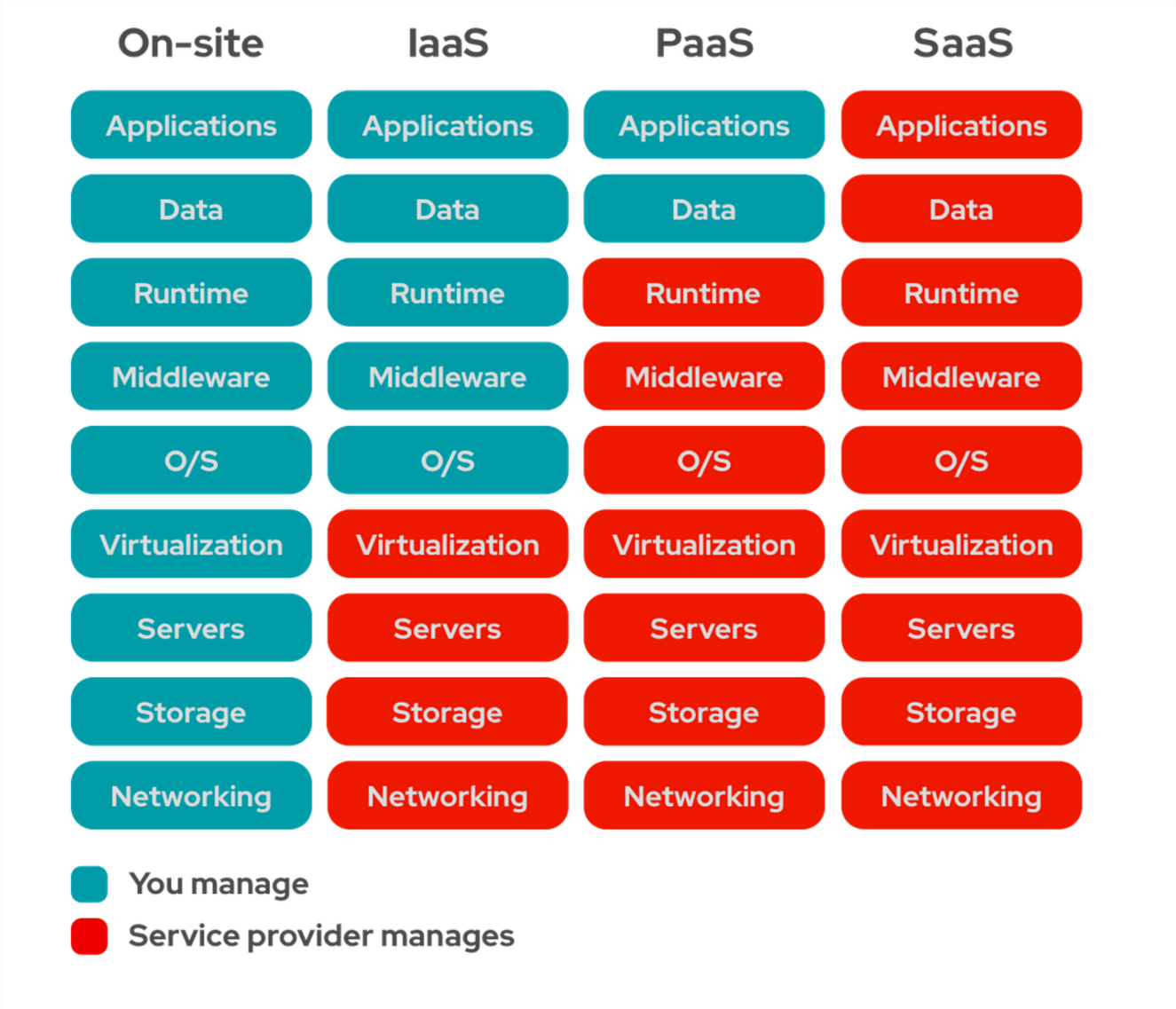
- IaaS - infrastructure as a service (VMs)
- PaaS - platform as a service (DB,k8s)
- SaaS - software as a service (applications)

Also:

CaaS = Container as a Service

FaaS = Function as a Service

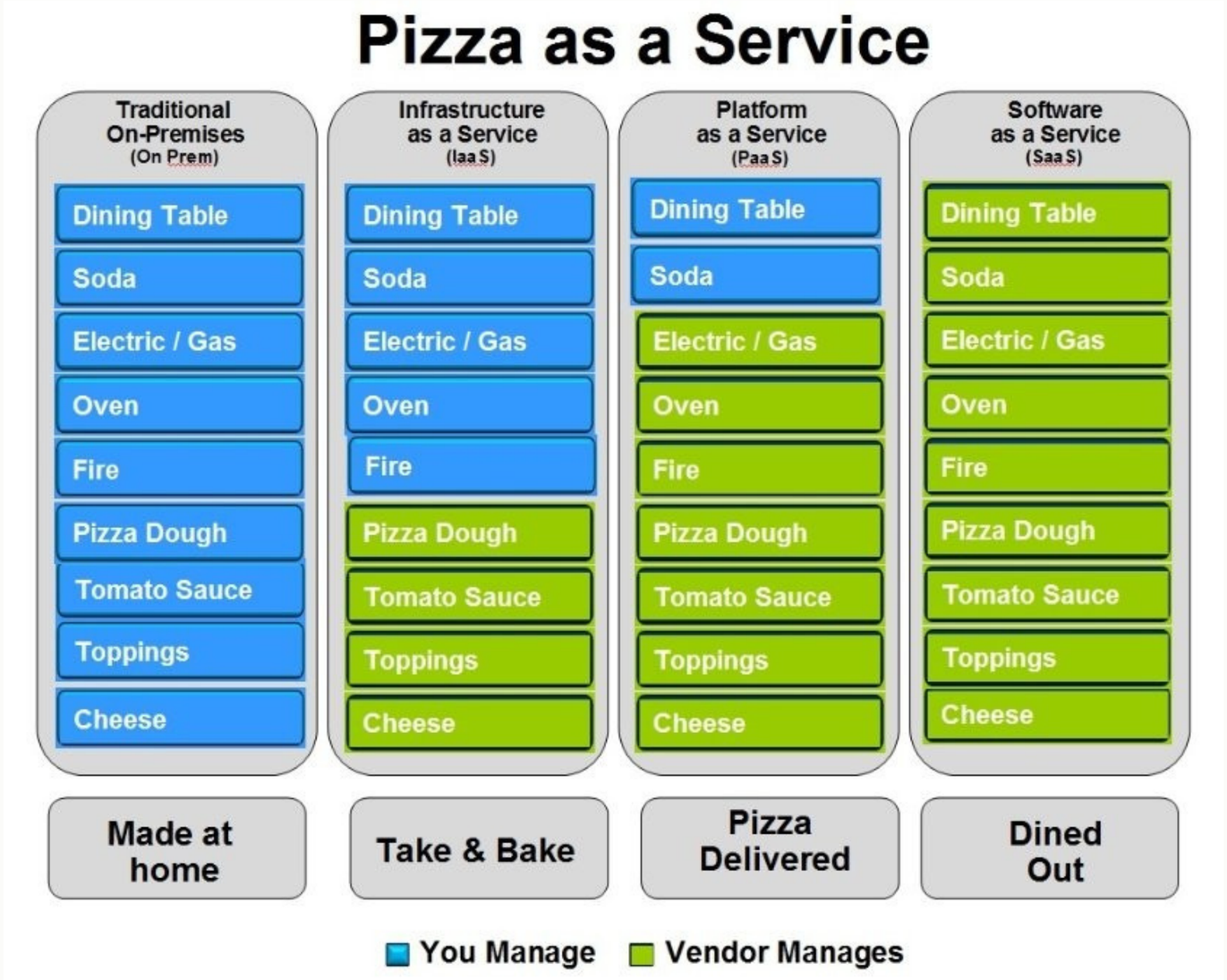
Cloud implementation models



Security, responsibilities and maintenance are shared between the cloud provider and the customer.

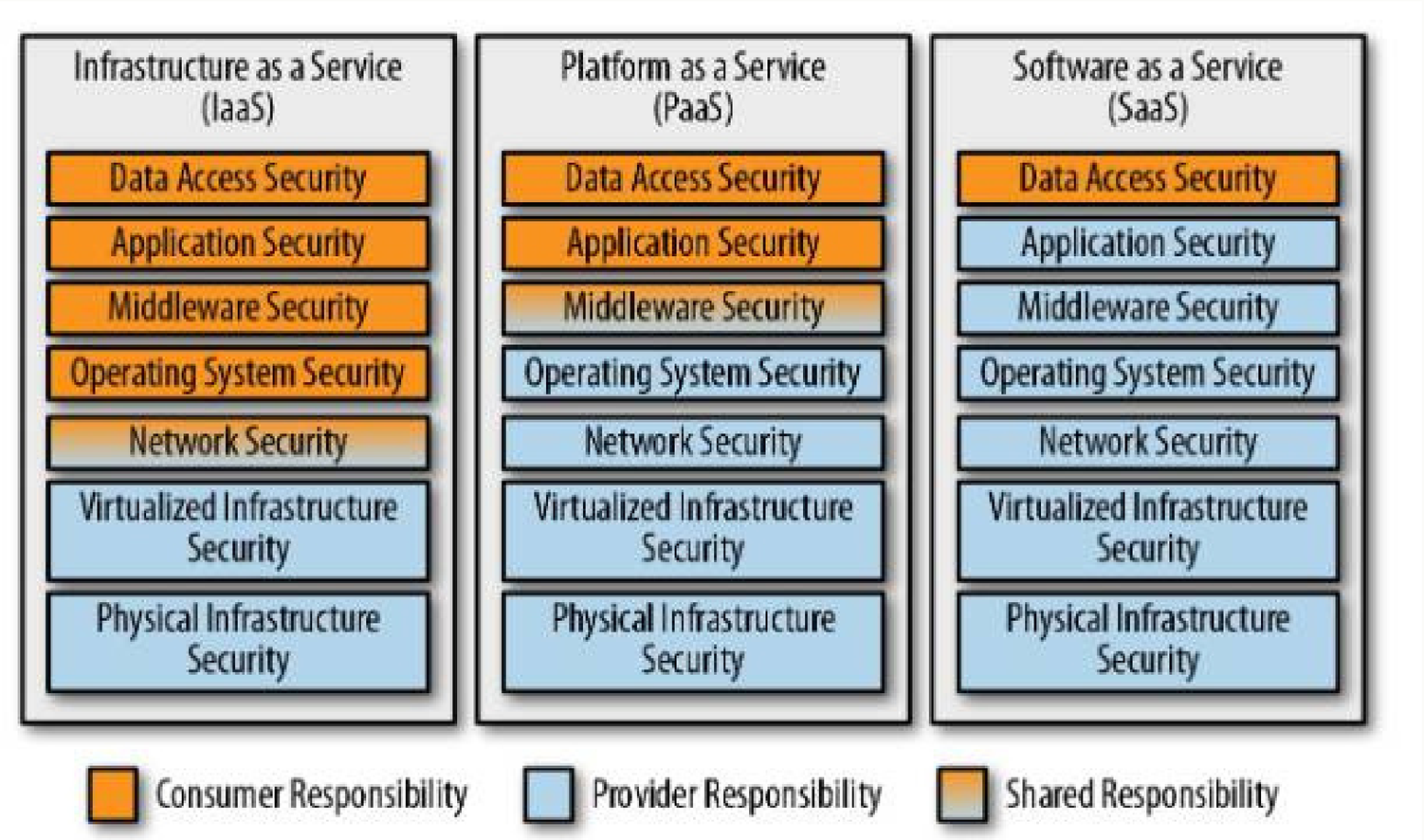
What is your responsibility? If you can touch it, it is your responsibility.

Pizza as a service



Source: Analogy by Albert Barron (Google)

Cloud models from security perspective



Source: Chris Dotson, Practical Cloud Security, O'Reilly

Security challenges in the cloud

- **for the 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; users are often in the role of sysadmin
- **for cloud providers**: complex network designs and no control over the state of VMs and services running on them

Common threats in the cloud

- cyber attacks: DoS, spoofing, man-in-the-middle,
- escalation of privileges, unauthorized access,
- hijacking accounts,
- misconfigurations of cloud resources,
- internal/external threats,
- malware,
- data breaches and unauthorised access,
- data loss,
- unpatched software and exploited vulnerabilities,
- inadequate encryption practices/implementation,
- insecure interfaces/APIs,
- external data sharing and data transfers,
- insufficient technical skills,
- VM escape,
- leaked credentials (published in git),

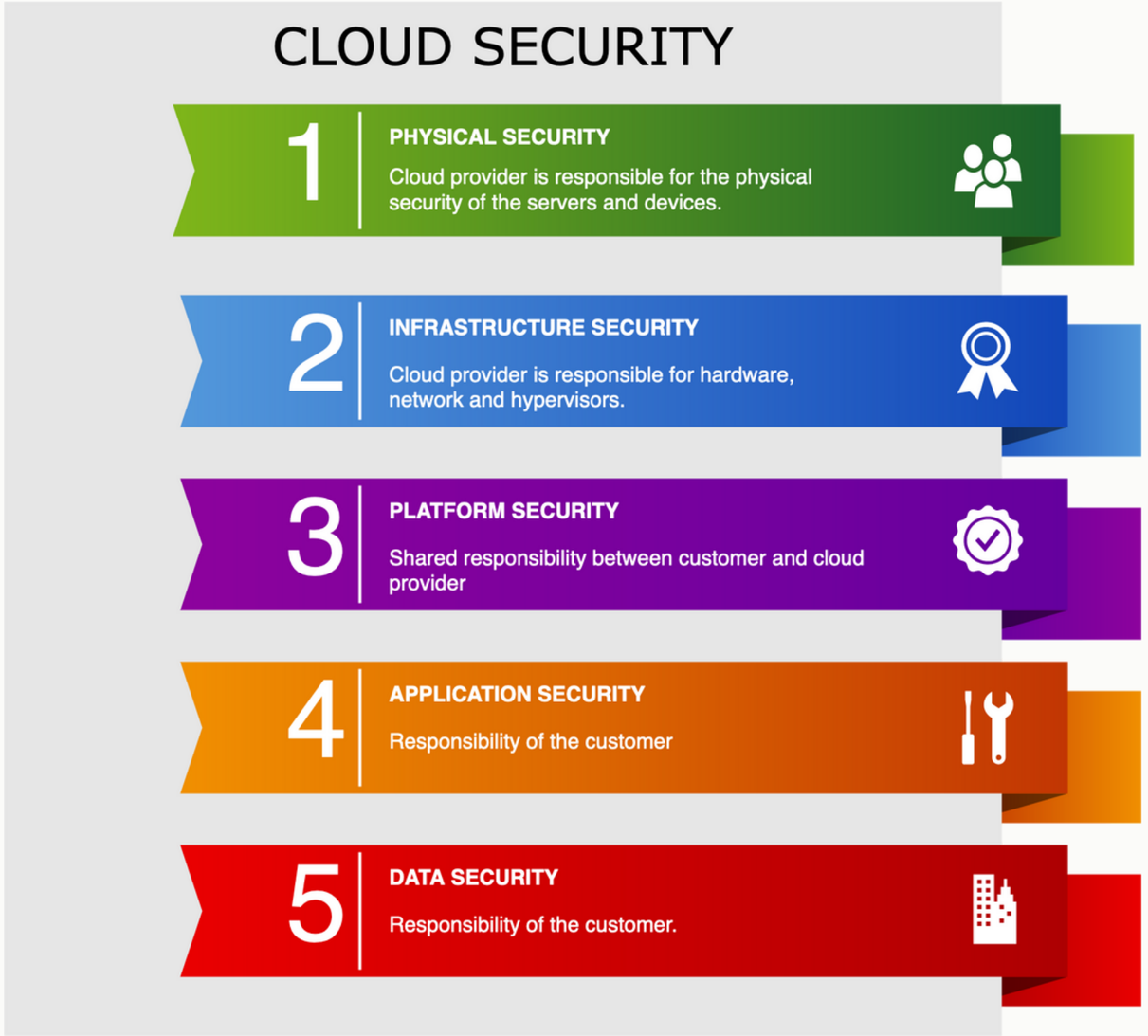
- misconfiguration and inadequate change control,
- lack of cloud security architecture and strategy,
- insufficient key management,
- limited cloud usage visibility.

How to prevent them?

To gain unauthorized access, attackers need to:
• gain access to VM or
• gain access to the host from the VM

- update your software regularly,
- apply MFA also for privileged accounts,
- limit access to the cloud services to a few locations,
- use encryption,
- apply network security,
- check VM for open ports, restrict access by enabling firewall,
- hypervisors need to be well protected and regularly updated.

Levels of security in the cloud



Fundamental cloud security concepts

- encryption,
- identity and access management,
- vendor lock-in,
- network security,
- virtualisation security,
- application security,
- data security.

Cloud data security

Cloud has multiple data stores:

- **object storage,**
- **block storage** and
- **file storage.**

- An object store is like a valet parking: you give a car to a valet, he parks and gives you a ticket, and you don't care where the car is parked (files as objects, the 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 a filesystem (NFS, CephFS)

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

Data security strategies

- Restrict access to data,
- use MFA,
- use tokenisation - replaces a piece of sensitive data with a randomly generated token
- use 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,
- monitor data-related activities, detect unusual events,
- to better collect, manage, analyze, and display log data, use SIEM.

Data encryption

- **States of data:**
  - data in motion/transit (being transmitted across network)
  - data in use (in memory, currently being processed)
  - data at rest (on persistent storage)
- encryption of data in use is relatively new, it has to be supported by the hardware (AMD SME, Intel SGX) - even the root user cannot read the data in memory, a processor can when that specific process is running.
- data at rest: encrypting files is fairly easy, but rarely done correctly, key management is a problem (use **HSM** = hardware security model, which is expensive, or **KSM** = key management service)

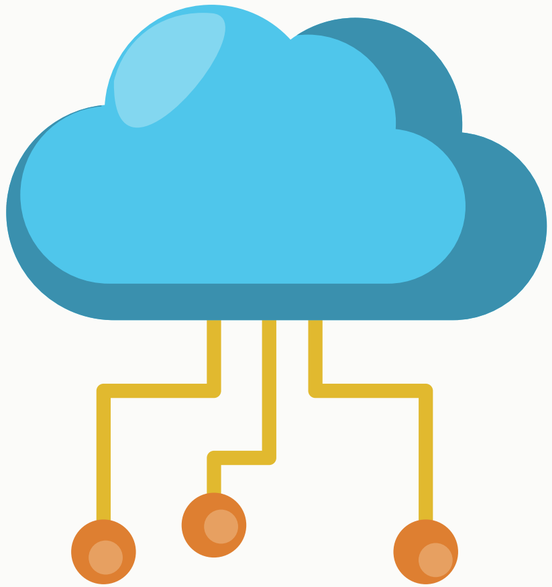
Private vs public cloud

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 are under the control of the organisation
- the potential of providing a perfectly safe environment (behind a firewall)

Public cloud:

- shared responsibility between the customer and cloud provider
- seemingly infinite resources
- the main target for security attacks (security is a big investment)
- no control over data for customer
- customer needs to trust the cloud provider



Private IaaS Cloud

Private cloud - IaaS

Security considerations

- use management network for core cloud services, a data network for storage nodes, APIs network for APIs, an external network for VMs and an 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,
- understand your security weaknesses,
- monitor your network,
- apply BIOS hardening and updates.

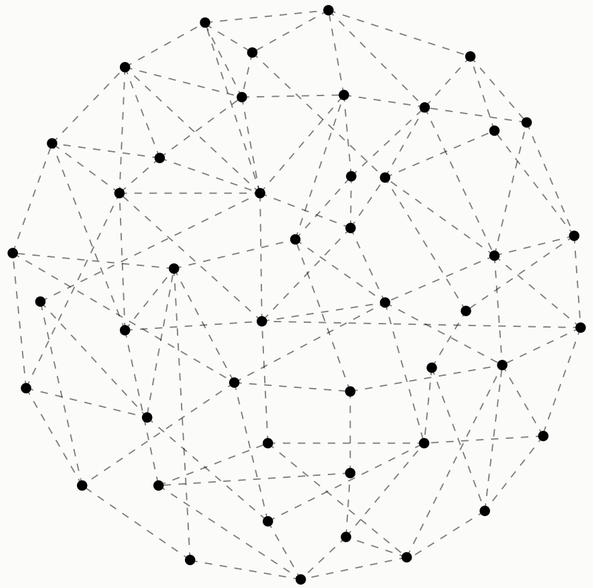
Private cloud - IaaS

Security considerations

- restrict access to IPMI/ILO interfaces, keep them updated, disable default accounts,
- security groups are not the same as ACLs, set allowed traffic per VM and name the security groups in such a 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 must be restricted (from certain IPs, by VPN, bastion host ..),
- outbound filtering with egress,
- use rate limiting on incoming connections.

Private cloud network security

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

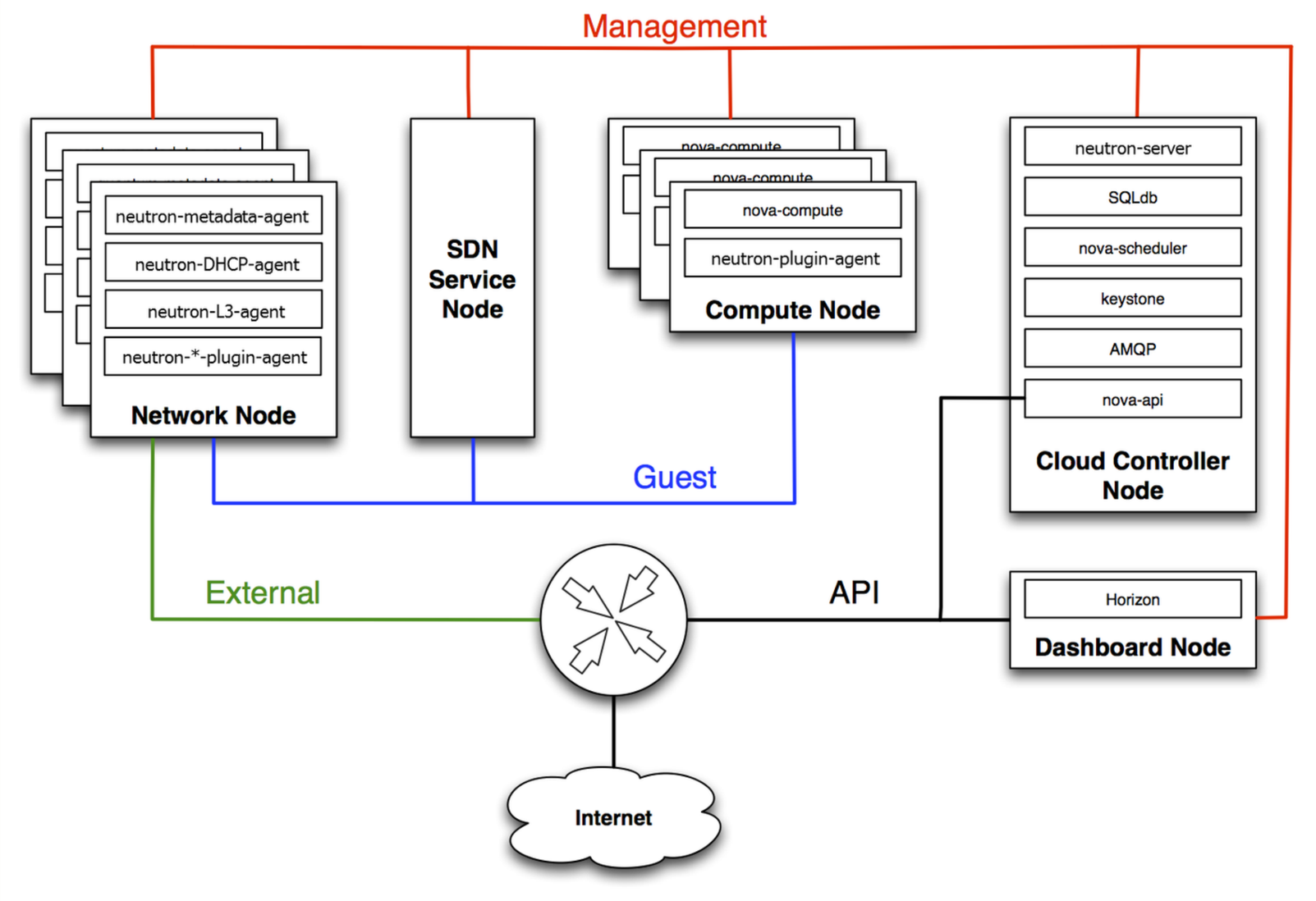


OPENSTACK NETWORK SEGMENTATION

OpenStack uses SDN, which complicates the design of the physical and virtual networks. There are typically 4 types of networks 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 topology



Fundamental network security measures

- **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 are implemented as a service, they apply per OS instead of per-network

Node provisioning

- use PXE boot for installations (automate node deployment),
- use a separate network for PXE,
- use automated configuration management,
- verify boot process (secure boot),
- use multitenant network,
- apply security measures already in the provisioning process.

APIs

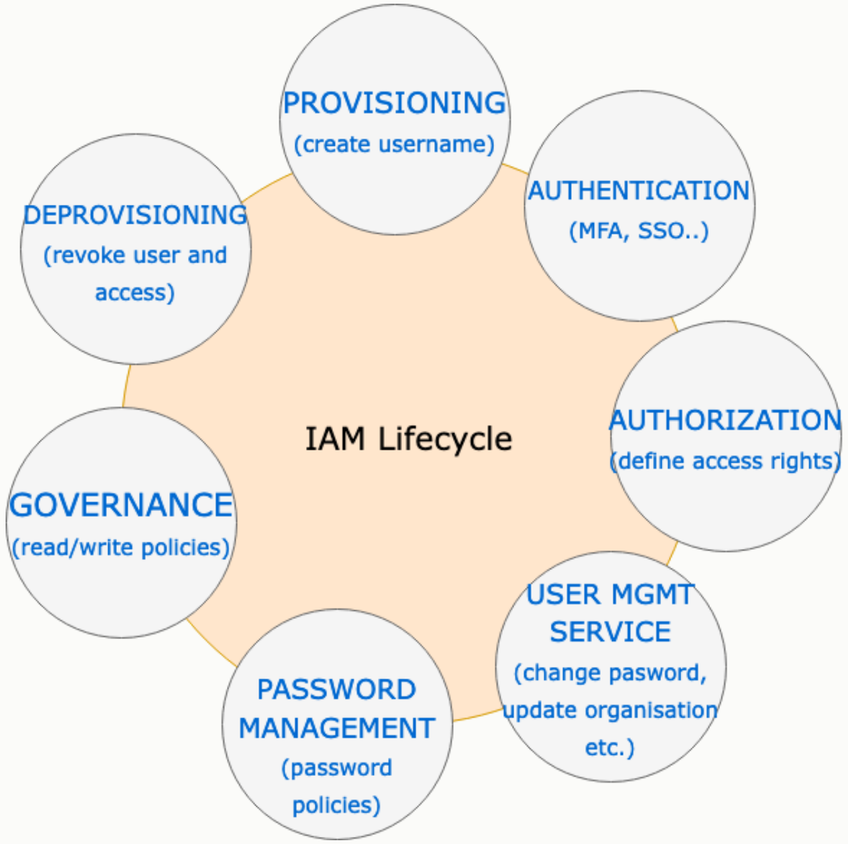
- APIs are the point of contact with the outside world, so a firewall as a security control is not enough
- usernames, passwords and tokens should never appear in the URL, use proxy caching and logging, data encryption
- isolate API endpoints on separate hosts
- use SELinux
- use host-based firewall rules
- use network ACLs and IDS
- use two-factor authentication where possible
- use a reverse proxy for REST API endpoints
- use WAF (Web application firewall to protect your APIs against common web exploits)
- OpenStack has private and public APIs, by default they are all public, but they don't have to be. Specify allowed API operations

Compute node security (Nova Compute)

- 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 page deduplication,
- use TLS for Spice/VNC sessions,
- don’t keep VM logs that could contain any sensitive data from the customer.

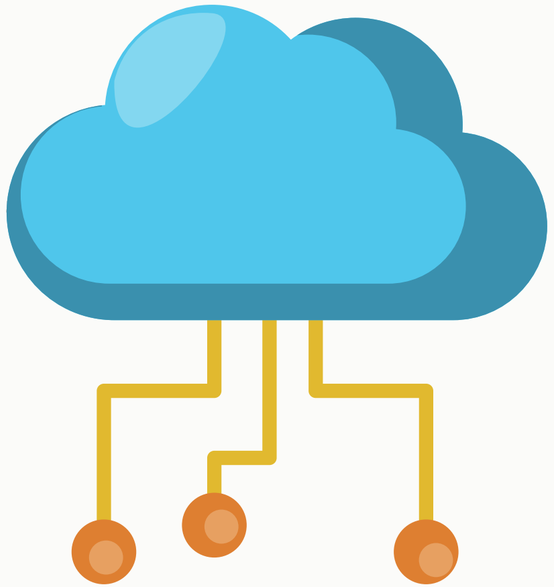
Authentication and access management

- authentication (identity) vs authorization (access)
- **traditional IT**: getting identity means obtaining email, VPN, and 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.



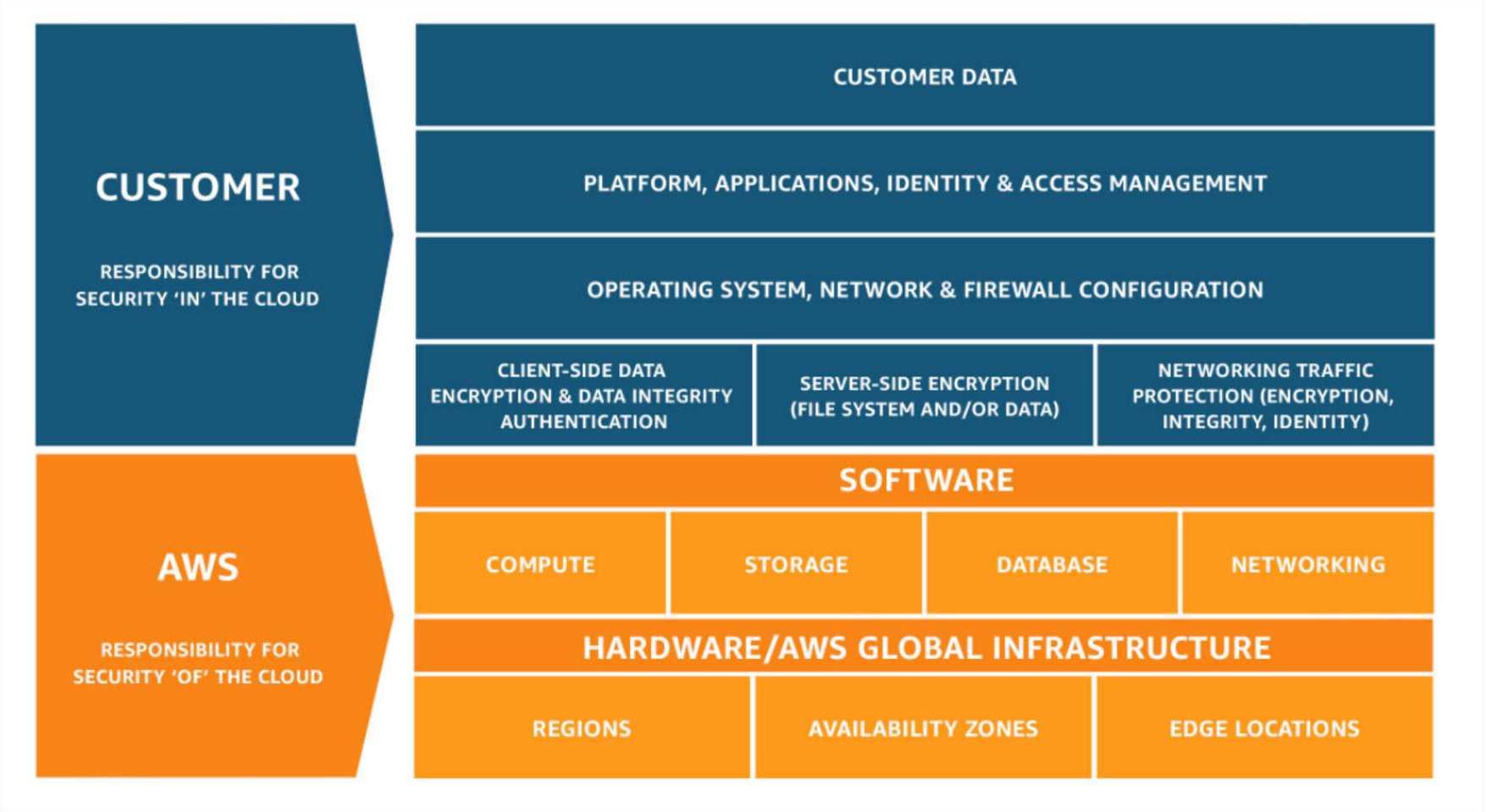
How to prevent common attacks?

- **Spoofing (faking someone's identity):** use SSH keys for authentication, TLS for communication, strong password policy, link Keystone with LDAP directory, MFA
- **Tampering (intentional change/corruption of data)**: use digital signatures for data integrity (Glance supports image signing), mandatory access control (MAC) and role-based access control (RBAC) to protect services
- **Repudation (denial of responsibility for an activity by a user or system)** : central logging and auditing in place, SIEM, monitor networks of anomalies (IDS/IPS)
- **Data disclosure**: use encryption, MAC/RBAC (role-based access control)
- **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



Public cloud

Security is shared responsibility



Fundamentals

1 **ENCRYPTION** - data encryption is not a problem, but key management implementation is complex and requires attention.

2 **IAM** - it is important to validate users, verify them, define roles and access rights

3 **NETWORK SECURITY -** consider network security groups, traffic inspection, software firewalls on VMs, geofencing, implementing zero trust.

4 **VIRTUALISATION SECURITY** where VM, containers and hypervisors need to be properly protected.

VENDOR LOCK-IN

Each vendor has its own proprietary settings and tools. They don't use the same encryption algorithms etc. It is problematic to transfer/export the data, migrate to another provider etc.

Infrastructure security in public cloud

SECURITY CONSIDERATIONS

- 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)
- automated creation of resources (Terraform/CloudFormation/Bicep) - the concept of Infrastructure as a Code
- 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

- automate network protection: self-defending network (threat intelligence, anomaly and intrusion detection (WAF))
- limit access (who can access what, from where, and for how long)
- perform vulnerability management
- automate the creation of resources (eg. AWS CloudFormation secure templates)
- automate configuration,
- provide OS hardening.

Public cloud security tools

- each public provider has its own set of security tools,
- even if tools are available, configuration is not trivial,
- hardening the cloud resources comes with a price.

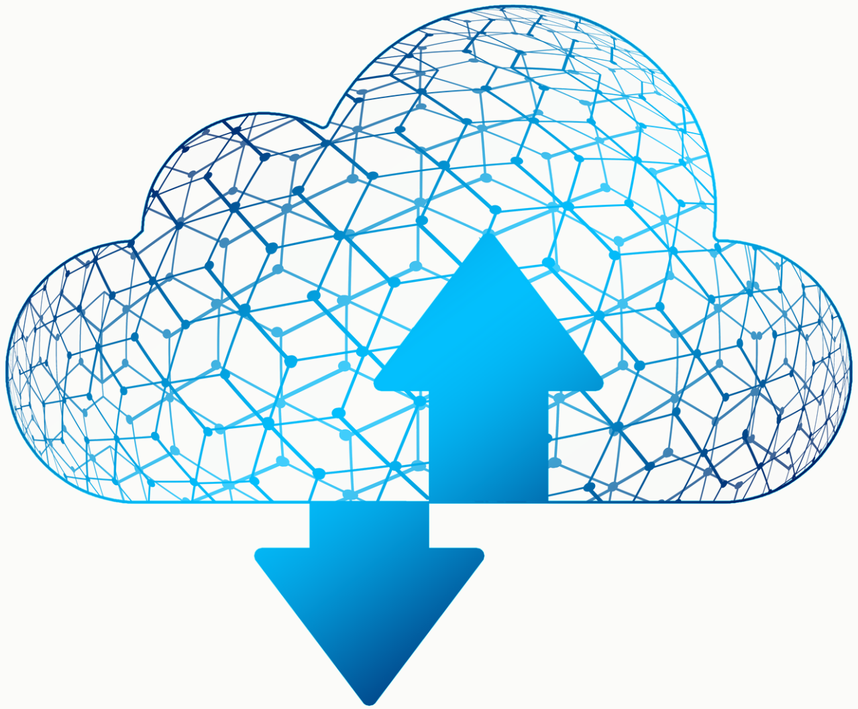
BUT:

- being a constant target of multiple attacks, public clouds are generally very secure,
- monitoring of events is already integrated into the cloud provider’s tools.

AWS SECURITY

- access policy: avoid using root, use MFA, disable credentials that are 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, and 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
  - **AWS Identity Manager** and **AWS organisations**

Zero trust and public cloud



Zero trust encompasses:

- explicit verification,
- least privileged access and
- constant assumption of breach. Think red, act blue.

- blocking access based on geo-location is not an implementation of zero trust, even though often offered as an option on public clouds,
- MFA can be implemented ineffectively,
- monitoring is often ineffective,
- secrets are often mishandled.

Why is MFA ineffective?

- legacy protocols are still used, such as FTP, POP3,
- MFA is not requested for all users (exceptions for privileged users),
- least privileged access is therefore not implemented sufficiently,
- using privileged accounts for daily operations increases the attack surface,
- access to IdP is not restricted,
- problem of token/cookie theft,
- secrets are not handled properly (credentials found in code (Github), in documentation),
- no encryption for transfers means travelling in plain text.

- Solution is IAM and no local accounts, centrally managed users and accounts, with short-lived credentials and access based on least privilege.
- Infrastructure as a code - automated creation of resources and orchestration (e.g. OpenTofu, Terraform..)
- To check IAM solutions of public cloud providers, follow this link: https://julian-wieg.medium.com/visualizing-multi-cloud-iam-concepts-63525967c0a7

Why is monitoring ineffective?

- not all native signals are monitored,
- data in SIEM is limited (cloud providers usually set a limit for the maximum number of events per second)
- log retention is very short,
- auditing is not enabled (in SaaS it is usually enabled to some extent by default)
- log visualisation makes sense if it is for a longer period of time, where anomalies are more apparent (logs are limited, storage is a cloud asset that one needs to pay for)

Bastion host problem

Bastion host allows access to the cloud environment

- host hardening should be in place,
- often insecure protocols are exposed to the internet (RDP),
- often protocols are poorly configured (e.g. SSH should have account lockout after multiple successive failed attempts, root login should be disabled, and password login as well),
- access to the host should be limited.

- Direct access to the public cloud is replaced by IAM - so no more accounts on the VMs, but short-term credentials to VMs via IAM.
- Users should be limited on the account level - e.g. AWS Service Control Policies.

Public 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 the cloud environment (autoscaling, serverless computers),
- customer shares responsibility with the provider for securing the 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 networks,
- use SIEM or threat detection solution,
- access should be limited with web application firewall, network ACLs, DoS protection.

APIs in public cloud

- in AWS Cloud all APIs are public by default
- Service Control Policies: where you make limitations for APIS,
  - whitelists,
  - limits to regions
  - etc.
- AWS API Gateway provides a number of ways to protect APIs from certain threats, like malicious users or bursts in traffic.
- You can protect your API by generating SSL certificates, configuring a WAF, setting throttling targets, and only allowing access to your API from a Virtual Private Cloud (VPC).

Data encryption and key management

- some public cloud providers provide HSM = hardware security model, which is expensive, or KSM = key management service) that is connected to an external HSM

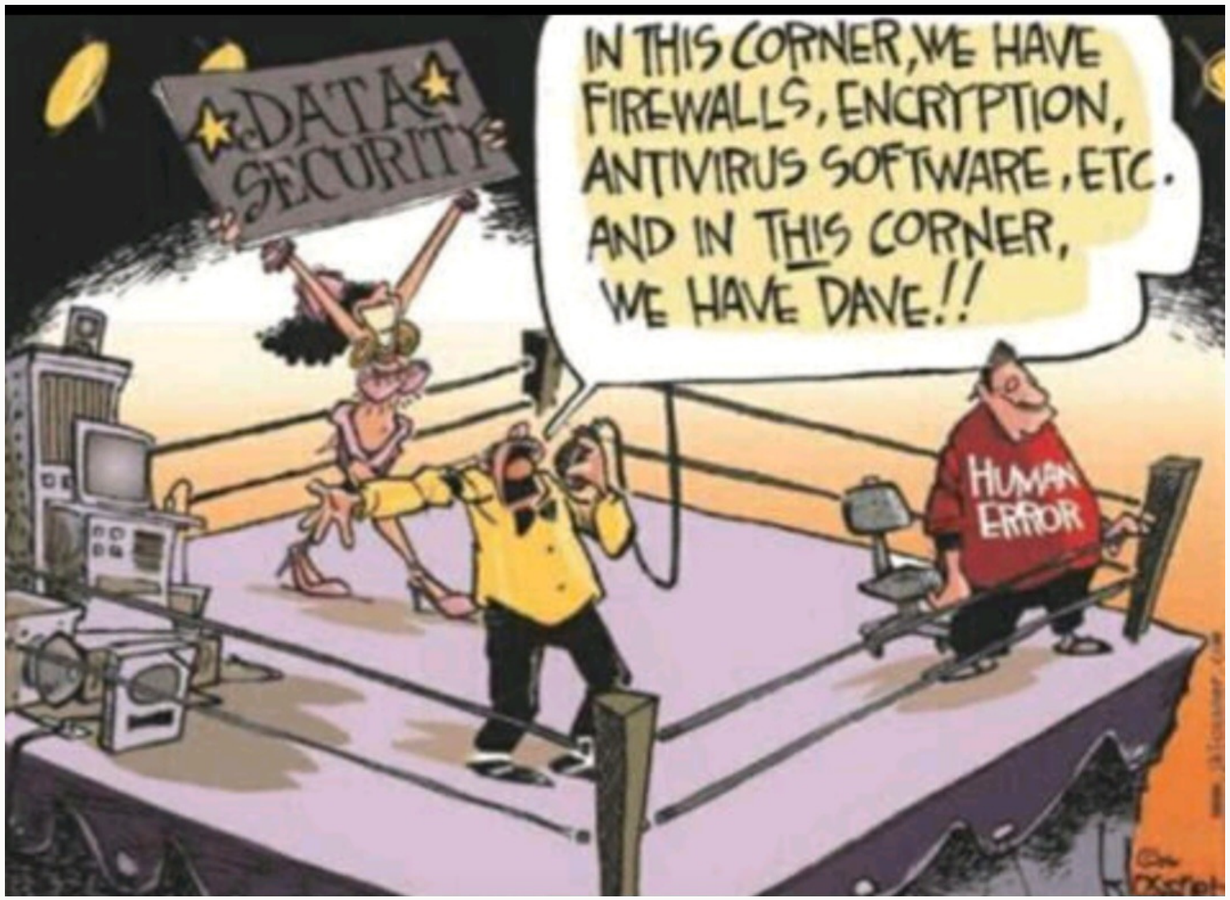
In 2021

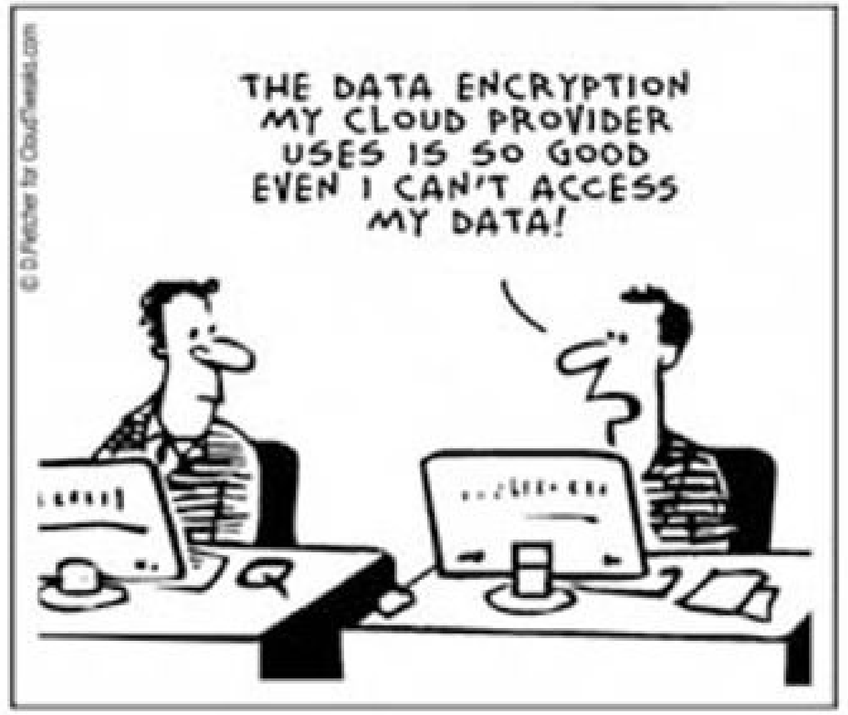
Provider	Dedicated HSM option	Key management service
Amazon Web Services	CloudHSM	Amazon KMS
Microsoft Azure	---	Key Vault (software keys)
Google Compute Platform	---	Cloud KMS
IBM Cloud	Cloud HSM	Key Protect

Today all public cloud providers offer HSM option for key management.

Source: Chris Dotson - Practical Cloud Security

Data security





Public cloud asset security

- assets in the cloud: VM, container, storage assets, images, network assets (Virtual Private Clouds (VPS), Content Delivery Network (CDN), DNS, load balancers, proxies etc.)
- each provider has its own provisioning method, they charge the provisioned assets
- due to the complexity of the cloud environment and lack of skills, provisioning should be done automatically - less opportunity for human error, the shift left approach can be used to apply security controls.
- **How to protect assets?**
  - Strong authentication and access control,
  - automated deployment with unit tests,
  - vulnerability scan, code tests etc.,
  - with continuous monitoring,
  - by placing the services into different network segments and
  - by having a disaster recovery plan in place.

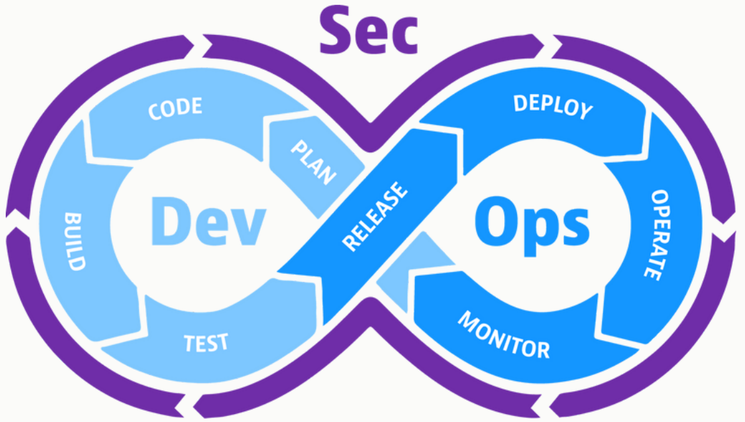
How to asses a public cloud provider?

- does it have security certifications?
- does it comply with relevant data protection regulations andlaws?
- evaluate the security controls they have in place, such as encryption, access controls, and authentication mechanisms,
- check incident response capabilities, read the SLAs and check how security is handled,
- check references, customer feedback, audit reports.

DEV-SEC-OPS

Shift-left approach to dev-sec-ops focuses on security measures in all steps of the development cycle, to ensure security already before the release of the software or launch of a cloud service. In other words, testing is moved to the left on the project timeline.

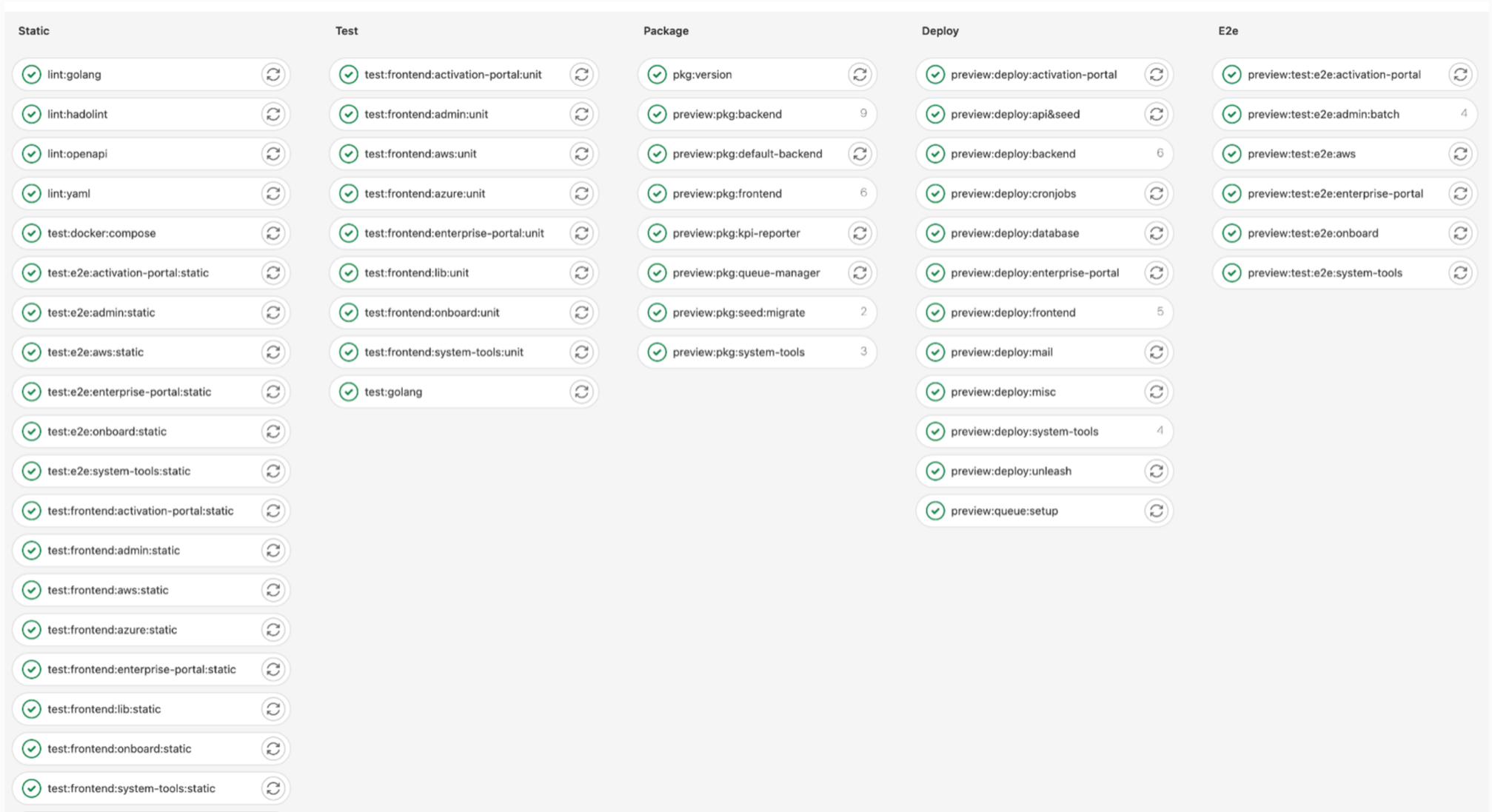
- provide unit tests (for cross-browser testing, APIs, integration)
- optimise test environments (use temporary environments),
- provide functional tests,
- check the quality of code,
- comply with OWASP security standards on secure coding,
- apply automatic vulnerability scanning,
- provide documentation and check it with CI
- provide SBOM (software bill of materials)



DEV-SEC-OPS TOOLS

- style checks (code, commits etc.),
- code quality and performance testing,
- compliance checks
- SAST - Static Application System Testing = scanning source code for vulnerabilities,
- DAST - Dynamic Application System Testing = black-boxing testing, application is exposed to different attacks,
- IAST - Interactive Application System Testing, hybrid of SAST and DAST, analyses applications with manual and automatic tests,
- RASP - Runtime Application Self Protection, software integrated to application to prevent attacks during runtime (blocking traffic for example) - tools like OpenRASP, Sqreen,
- secret detection - scan code for secrets,
- dependency scanning for security issues - tools like WhiteSource, FOSSA,
- application hardening (WAF),
- security monitoring.

Example



3fs company

Quiz

1. Who’s responsibility is the network security in IaaS cloud?
2. What is the difference between virtualisation and cloud?
3. Which is the main problem with encryption?
4. Name one attack that is related to encryption.
5. Why is MFA often ineffective?
6. How can we protect data in the cloud?
7. What is a security concern when considering implementing software-defined networking (SDN)?
    - A. It has a decentralized architecture.
    - B. It increases the attack footprint.
    - C. It uses open source protocols.
    - D. It is cloud based.
8. Which security measures can prevent privilege escalation in the cloud?
9. How to protect development process?
10. Is the following true or false:

- Obfuscation: The deforming of code to such a degree that even if the source code is obtained, it is not easily decipherable.
- Masking: A weak form of confidentiality assurance that replaces the original information with asterisks or X's.
- Data classification entails analyzing the data that the organization retains, determining its importance and value, and then assigning it to a category.