Virtualization

SecDev - Fall 2019

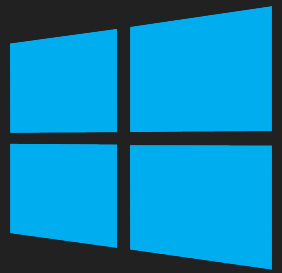
Operating System

-Software

-Controls communication with hardware

-Launches and manages applications

-Handles I/O Peripherals



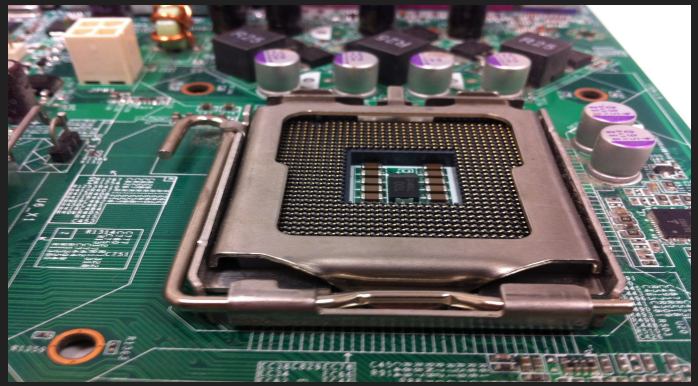
macOS

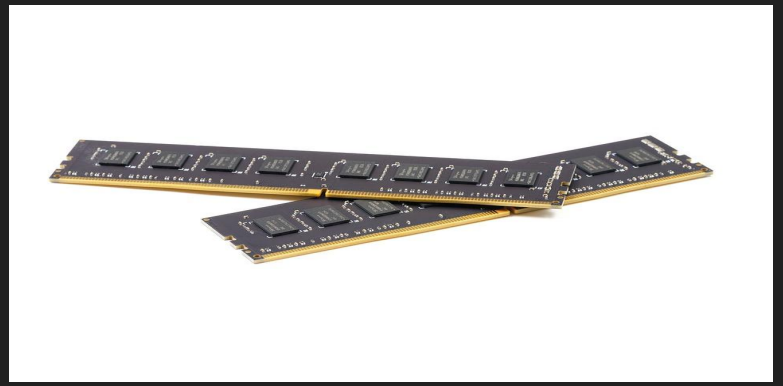
Hardware

-Underlying physical components

-Directed by software

-Easy to swap out and upgrade

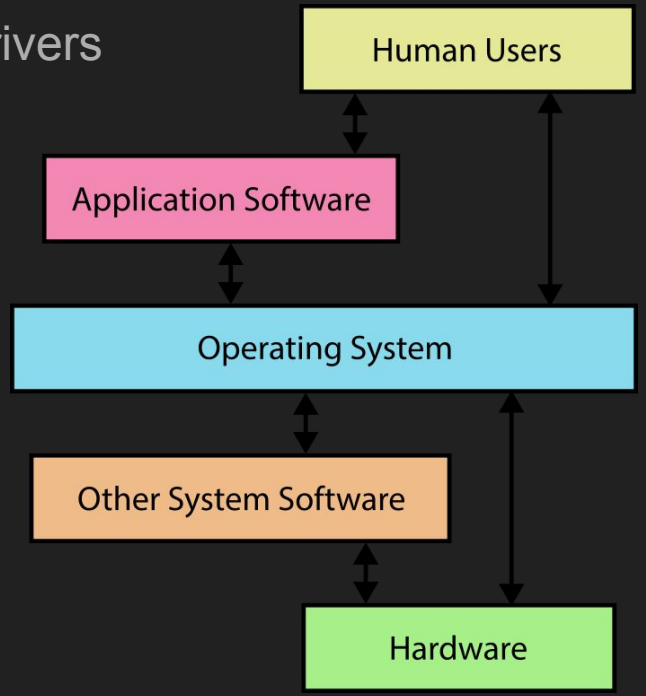




Typical Computer

-A typical computer has one operating system installed directly on top of hardware.

-Operating system links to hardware through specific drivers



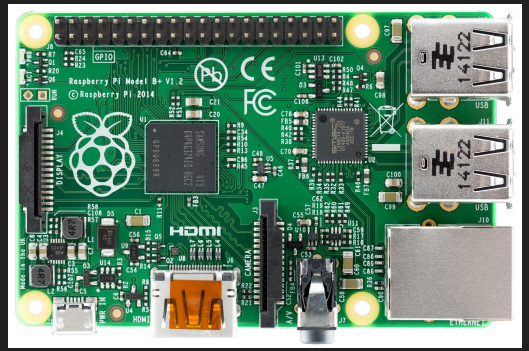
CPU

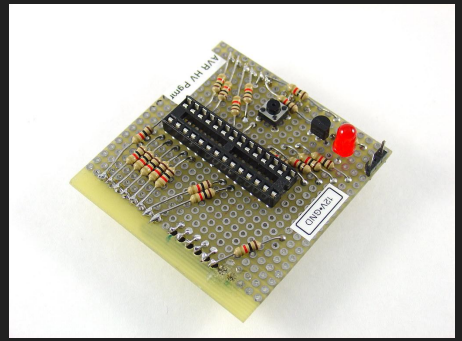
-Central Processing Unit

-Executes Code

-CPUs vary in speed and ISA (Instruction Set Architecture)

-x86, RISC-V, AVR and ARM





RAM

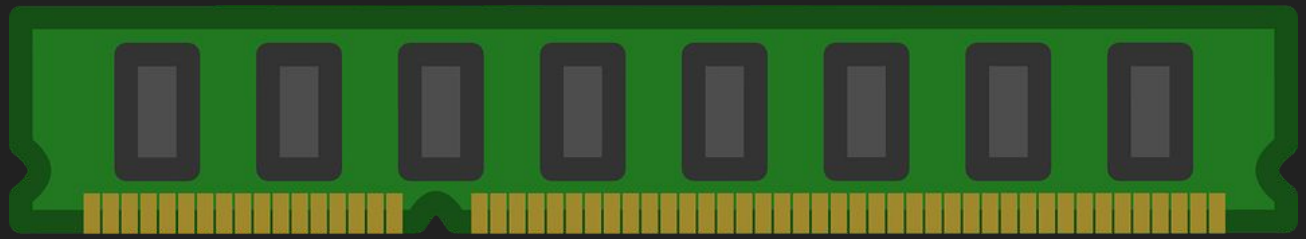
-System memory

-Used by OS for working applications

-More is better

-Does not store information, data loss on power off

-Faster than disk but slower than CPU cache



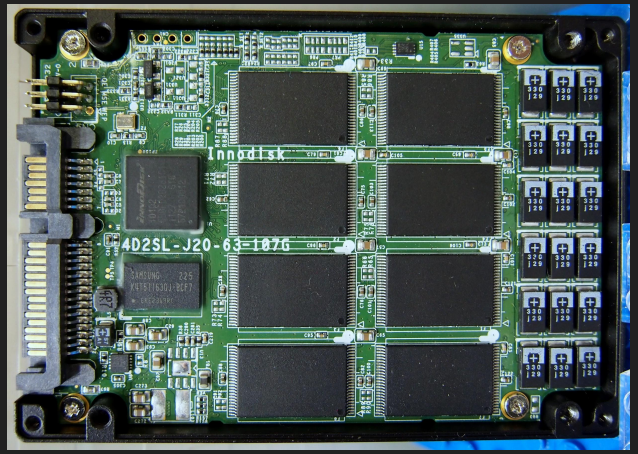
Disk

-Permanent storage

-Slower than RAM

-Where OS, applications and files typically are stored

-HDD, SSD, Flash



I/O

-Provides means for humans to interact with computers

-Throughput is limited by means of interaction

-Mice, keyboard, video monitor, printer, CD drive

Vocab

Virtual Machine - a software computer comprised of configuration files and backed by the physical resources of a host

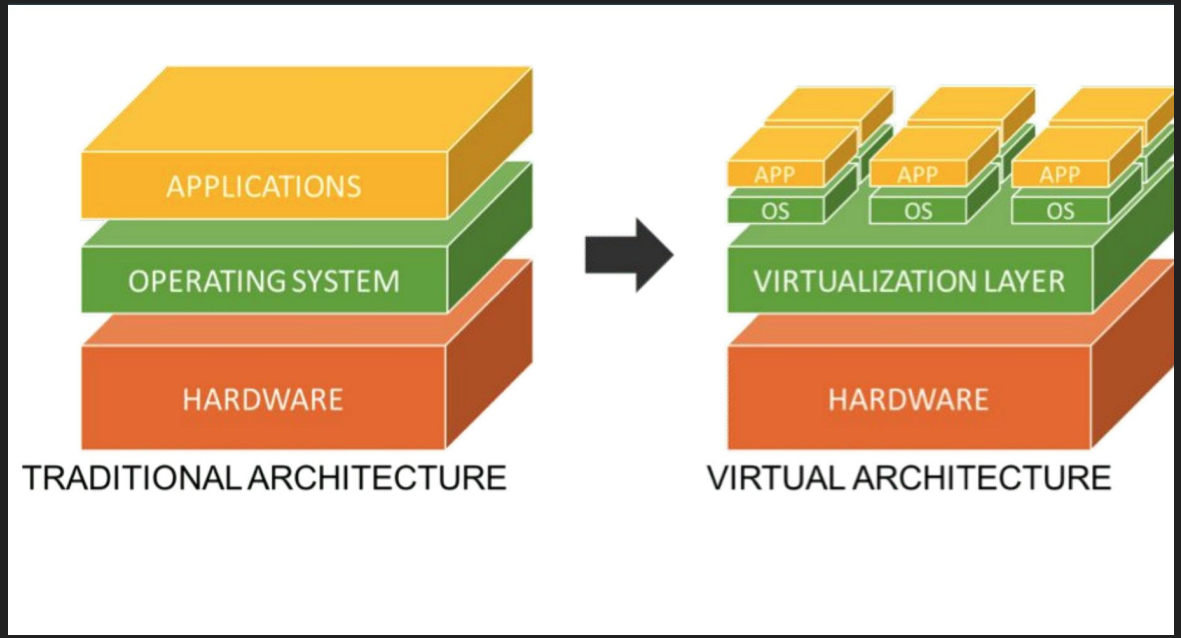
Hypervisor - A Hypervisor is an OS that separates a computer's operating system and applications from the underlying physical hardware

Host System - OS installed on physical hardware

Guest System - Virtualized OS on top of Host System

How do they compare?

-A hypervisor manages multiple operating systems on top of a single or multiple pieces of hardware



How do they relate?

-When configuring a virtual machine you have the opportunity to decide what hardware the guest operating system has access too

-Understanding how the physical hardware impacts the performance of the virtual machine

-Configuration of virtual I/O is important to prevent conflicts (mouse input, processor states, etc)

Scenario

-Suppose we are a small business that uses a web application. To host our online business we would need a web server and a database.

-Typically this means buying two servers one for web and one for the database.

-Operating systems are installed directly onto hardware

What’s the worst that could happen?

-Discuss briefly why this could lead to problems down the road...

Well...

-Operating system gets linked to exact hardware via drivers

    -Hardware failure leads to broken operating system

-During off-peak hours, system resources sit idle

-If a virus is found the whole operating system must be re-installed or installed from backup

-OS and applications must be updated regularly

How can virtualization help?

-Separates hardware from operating system

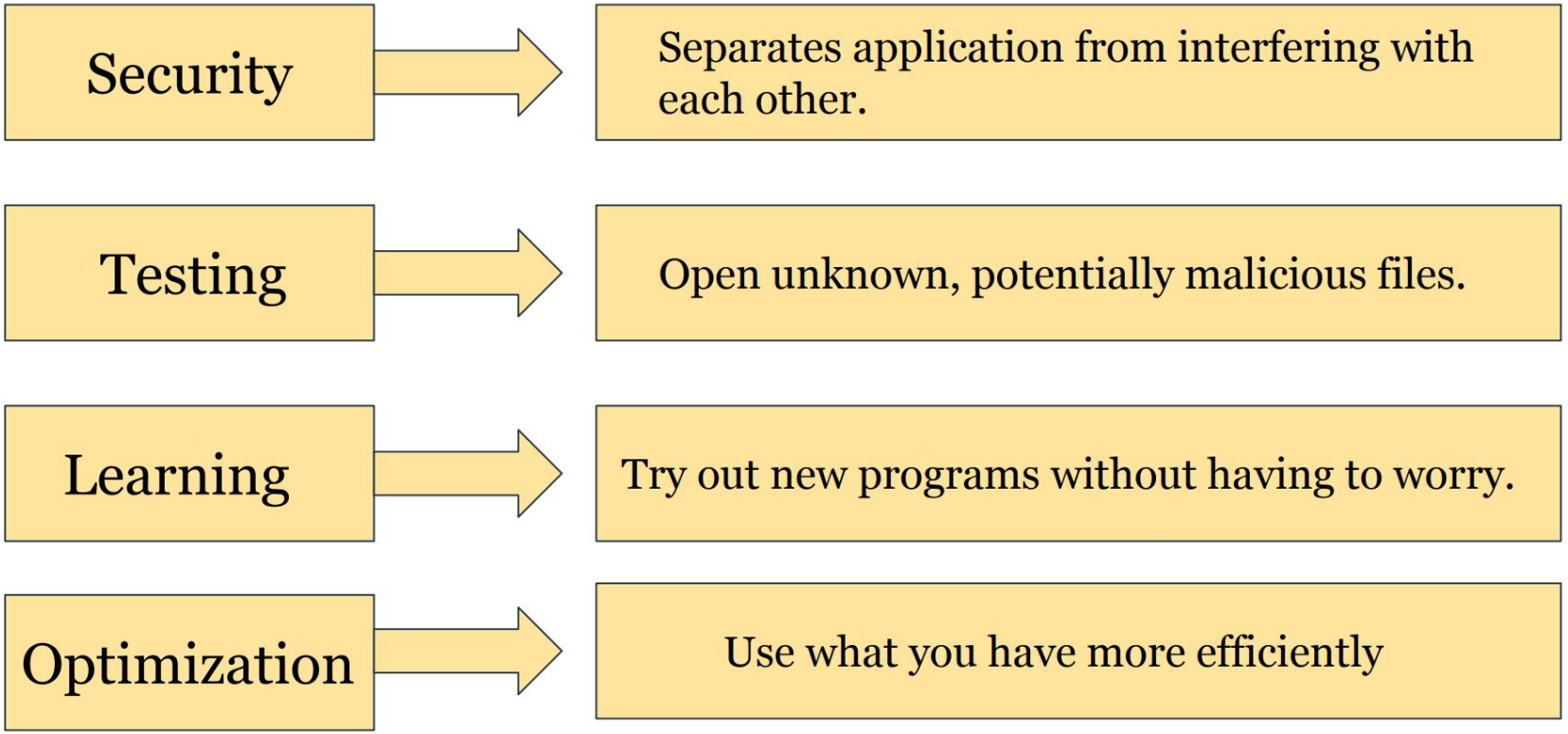
-OS is treated like a file

-Snapshots can be regularly taken in comparison to backups

-Lower initial cost and operating cost

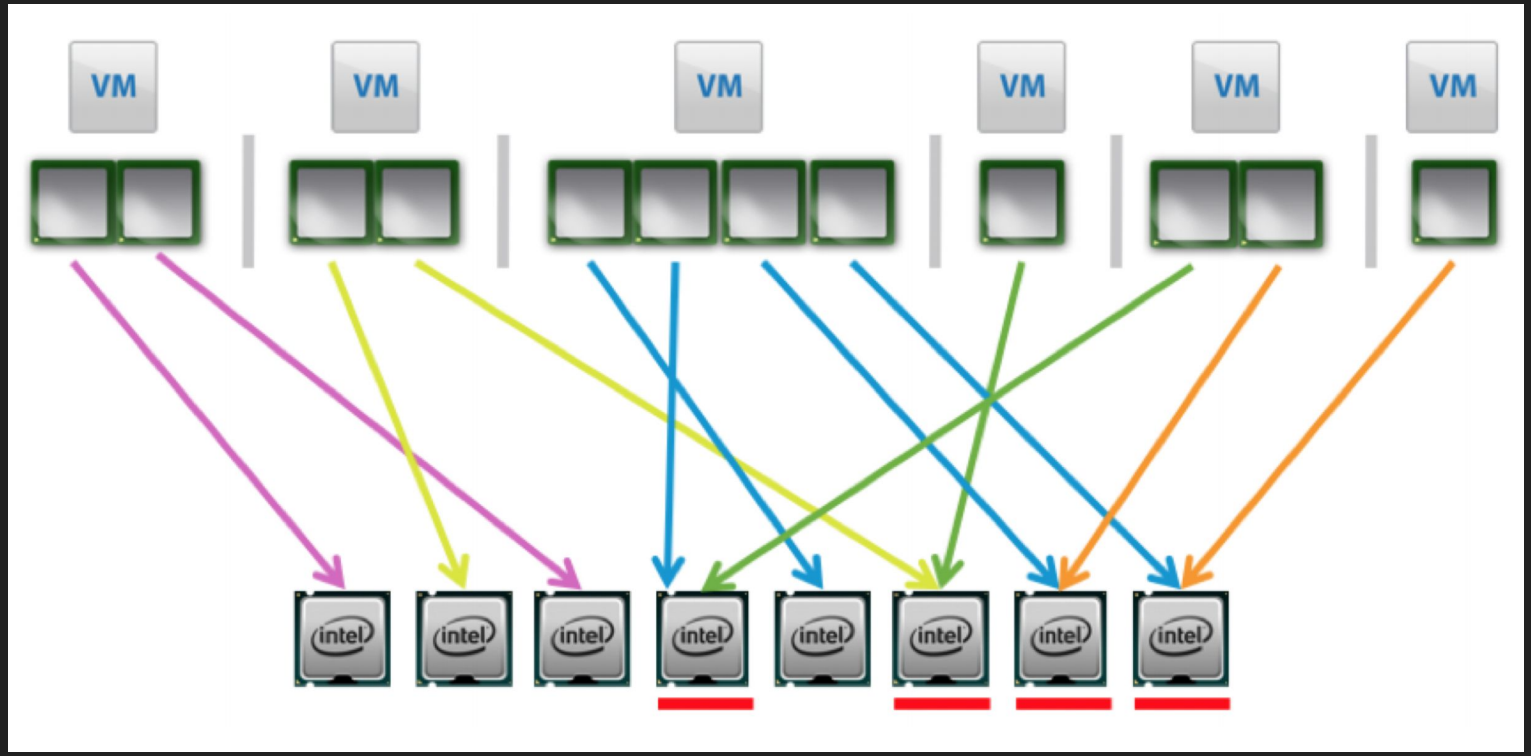
-Easier provisioning of resources

What are the benefits of Virtualization?



Characteristics of Virtual Machines

- **Partitioning**
  - Run multiple operating systems on one physical machine.
  - Divide system resources between virtual machines.
- **Isolation**
  - Provide fault and security isolation at the hardware level.
  - Preserve performance with advanced resource controls.
- **Encapsulation**
  - Save the entire state of a virtual machine to files.
  - Move and copy virtual machines as easily as moving and copying files.
- **Hardware Independence**
  - Provision or migrate any virtual machine to any physical server.


VM
VM
VM
VM
VM
VM
intel
intel
intel
intel
intel
intel
intel
intel

Types of Virtualization

-There are two types of virtualization that you can run

-Each has their own benefits/drawbacks with varying use cases

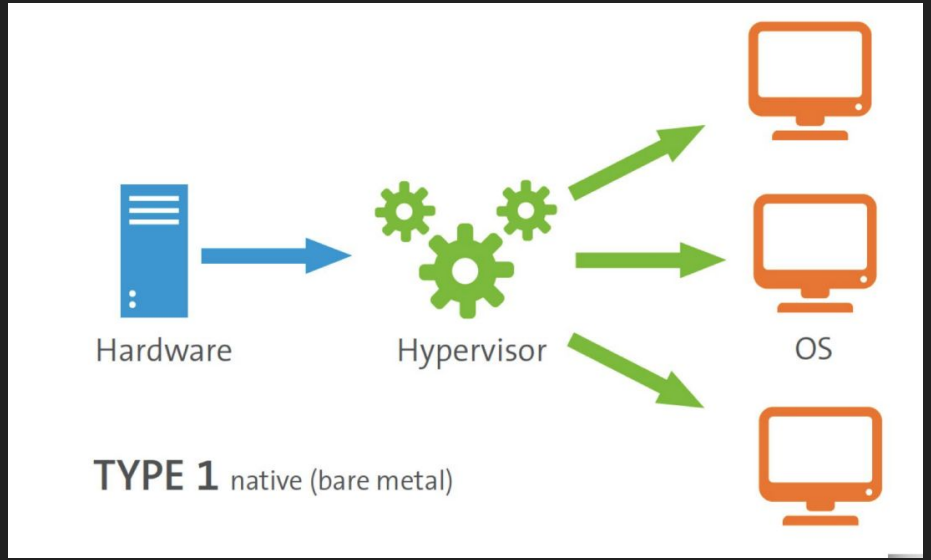
Type 1

-Hypervisor on bare metal

-Higher performance

-Typically for large racks of servers

-VMware ESX/ESXi, Hyper-V, Xen



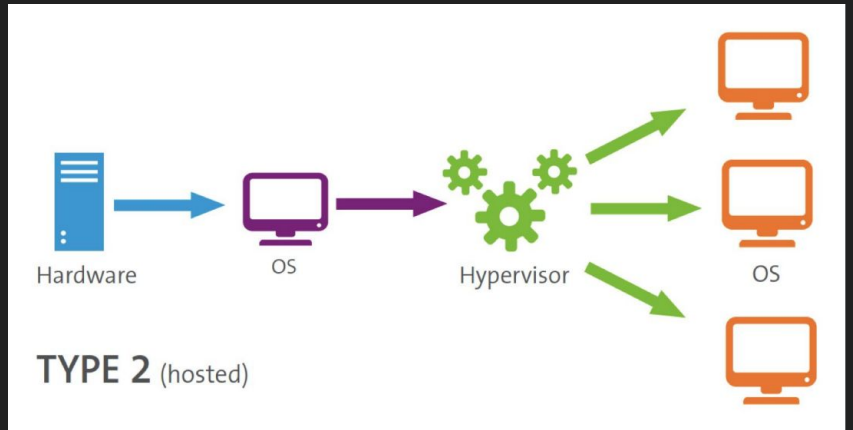
Type 2

-Hypervisor on an Operating System

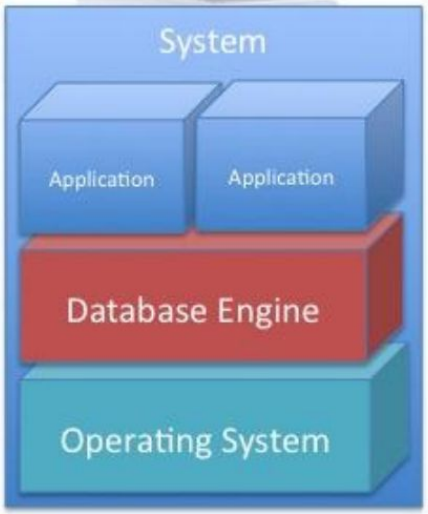
-More overhead, less performance

-For Desktops and Laptops

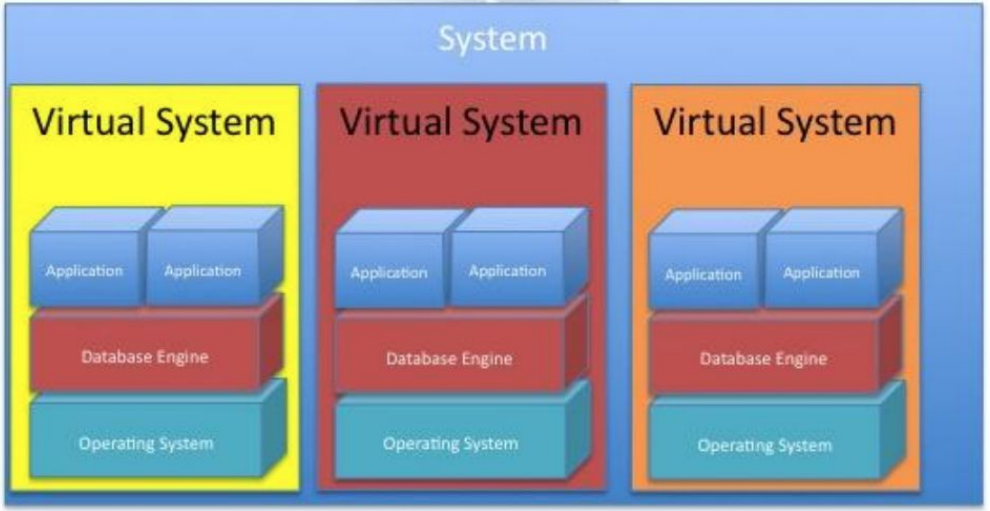
-VirtualBox, VMware Workstation/Fusion, KVM, parallels



Operating System and Applications on a Physical System



Virtual Systems Running under a Type 1 Hypervisor



Nested Virtualization

-The ability to run a virtual machine within another

-One or more hypervisors inside another hypervisor

Para-virtualization

-Guest apps are executed in their own isolated domains, as if they are running on a separate system, but a hardware environment is not simulated.

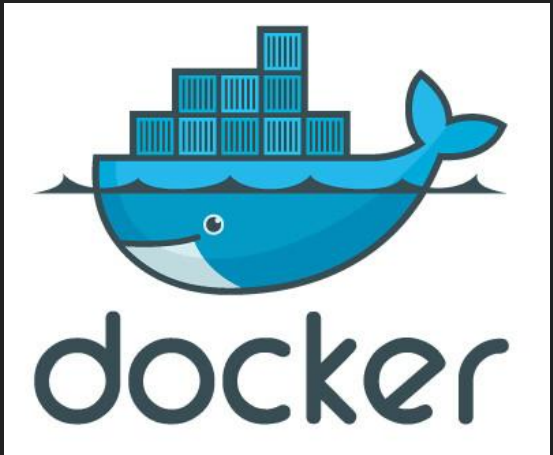
-Guest programs need to be specifically modified to run in this environment

-Unlike virtualization no hardware is simulated and only OSs are managed

Docker

-Docker is a popular paravirtualization tool

-Builds portable containers that can deploy anywhere



Difference?

-Performance benefits

-No hypervisor, much more stable without the need of more resources

-Predictable app behavior within the container

-Distribution ensures anything downloaded works the exact same within the container

-Developers get exactly what they need, no more no less, only dependencies that are needed get downloaded

Difference cont.

-Containers are smaller on disk

-Containers can be packaged, making them more portable

Container Orchestration

-Kubernetes, AWS ECS/EKS

-Tool for mass deployment, scaling and management

-Containers as code

kubernetes

Issues?

-No isolation from host OS, container tool runs as application to host OS

-Container malware can be distributed much easier between containers and possible infect the container application

-Denial of service attacks are possible if multiple containers are running within one host

-Multiple containers share same kernel resources, if you grab kernel resources, you can starve all the containers on the system

Emulation

-Mimicking specific hardware that a game expects to run on

-Either translating the expected CPU instructions to actual or virtualizing actual hardware wire for wire

break?

-if its been ~an hour

Cloud Computing

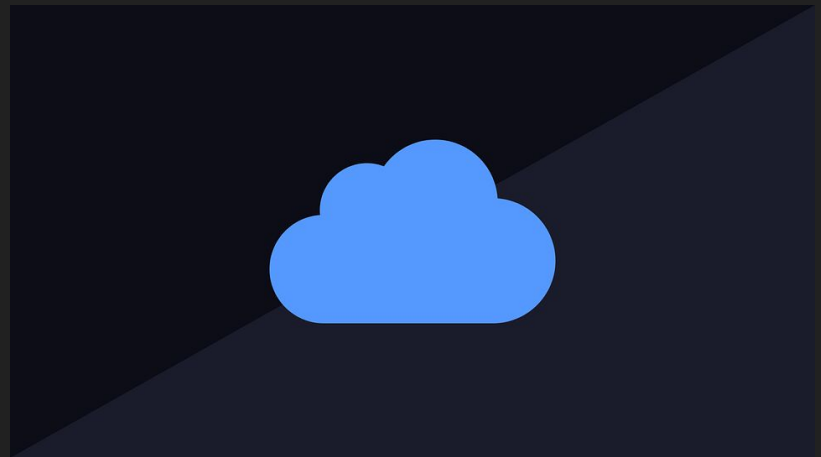
-”On-Demand Computing”

-Available computer resources(Compute/Storage) that users do not have to actively manage

-Large data-centers that users can interact with over the internet

-”Pay as you go”

-Popularized by Amazon EC2



Vocab

Public cloud - Network is open for public use and is managed by a private organization (AWS)

Private cloud - Owned by an organization, self-run data-centers, for use by organization only

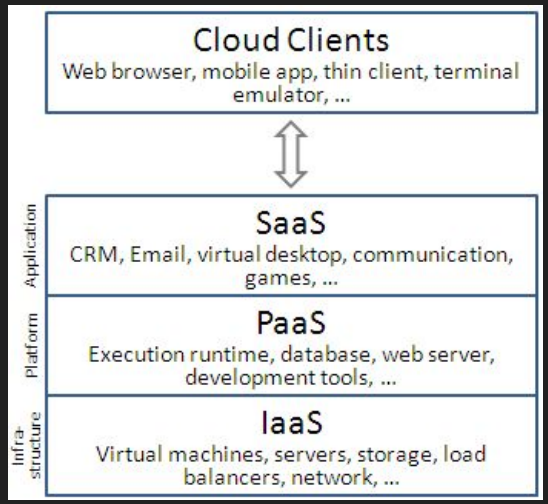
Hybrid cloud - usually a private organization that has their own private cloud, but also makes use of public cloud systems

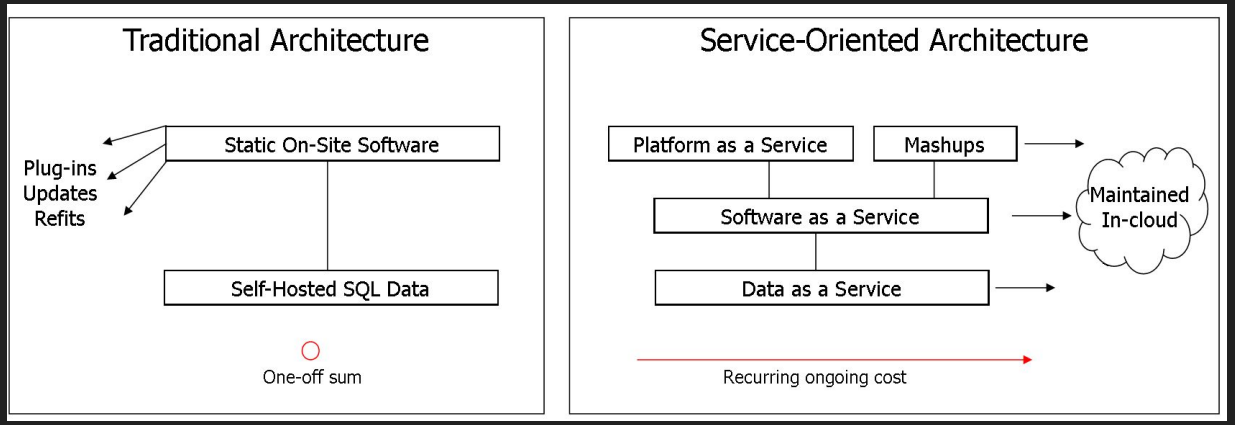
Service Oriented Architecture

-”Everything as a service”

-Providers offer services according to different models

-Different models offer different layers of abstraction





Infrastructure as a Service (IaaS)

-Xen, Oracle VirtualBox, KVM, VMware ESX/ESXi, Hyper-V

-Alternate to hypervisors are Linux containers, which run in isolated partitions of a single linux kernel (running directly on physical hardware)

-Virtual Machines, Virtual Storage, Virtual Networks

Cloud Orchestration

-Computers as code

-Ansible, Puppet, Terraform, AWS CloudFormation

Platform as a Service (PaaS)

-More abstract than infrastructure as a service

-Execution runtime, database, web server

-Provider offers and environment to build off of (Languages, Libraries, Services)

Software as a Service

-Typically accessed with a thin client and web browser

-Only running the code you need

-Games, Email, Communications

Network as a Service (NaaS)

-Provisions virtual networks on the providers network infrastructure

Server-less computing

-Cloud computing execution model in which the cloud provider fully manages requests to run code

-Requests are billed by how much resources they use

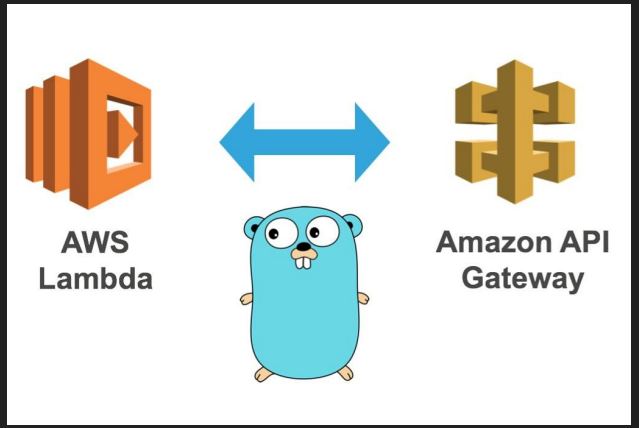
-Despite the name, it does not actually involve running code without servers

Function as a Service (FaaS)

-Procedure calls that leverages serverless computing to run single or chained functions in the response to events

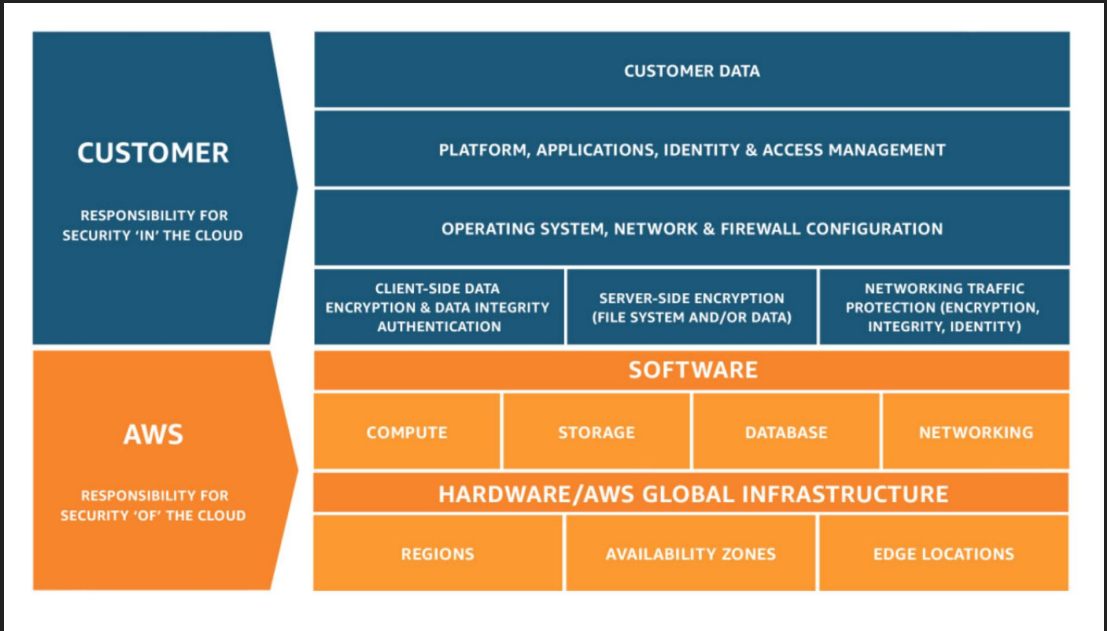
-AWS Lambda is the most popular type of FaaS

-Often used in conjunction with cloud monitoring tools



Shared Responsibility

-As you increase abstraction in the cloud, more and more responsibility for the security of the application will shift towards the cloud provider



Security example

-When running services in the cloud, you have the ability to automate high volume low priority tasks

-FaaS can be used to automate…

- Updating packages
- Auto-remediate open security groups
- Noticed an intrusion, auto remediate with step functions and lambda

Security in the Cloud

-Access data at any time on the public cloud

-Accidental alteration or deletion of information by cloud provider or users

-Cloud provider can share information with third-parties or law without warrant

-Encryption is recommended to protect sensitive data

-Access is managed with “Identity Management System” (AWS IAM)

Security Cont.

-Shared responsibility model allows developers to only worry about the security of their code and not the underlying backend

-There are security tools developed specifically for the cloud. AWS Cloudwatch, AWS Inspector, AWS Guard-duty, AWS Security Hub

-Everything is code. Easier to detect break-ins and unusual activity. Easier to develop tools that manipulate and monitor services because its all API requests

Cloud Security Alliance - Top three threats

-Insecure Interfaces and APIs

-Data Loss and Leakage

-Hardware Failure

Limitations and Disadvantages

-Limited customization options

-Fewer options at a much cheaper price

-Legal Limitations