- Cloud Models
    - Infrastructure as a service (IaaS)
        - Sometimes called Hardware as a Service (HaaS)
            - Outsource your equipment
    - You're still responsible for the management
        - And for the security
    - Your data is out there, but more within your control
    - Web server providers
- Software as a Service (SaaS)
    - On demand software
        - No local installation
        - Why manage your own email distribution or payroll?
    - Central management of data and applications
        - Your data is out there
    - A complete application offering
        - No development work required
        - Google Mail
- Platform as a service (PaaS)
    - No servers, no software, no maintenance team, no HVAC
        - Someone else handles the platform, you handle the development
    - You dont have direct control of the data, people, or infrastructure
        - Trained security professionals are watching your stuff
            - Choose carefully
    - Put the building blocks together
        - Develop your app from what's available on the platform
        - SalesForce.com
- Anything as a Service
    - A broad description of all cloud models
        - Use any combination of the cloud
    - Services delivered over the Internet
        - Not locally hosted or managed
    - Flexible consumption model
        - No large upfront costs or ongoing license
    - IT becomes more of an operating model
        - And less of a cost center model
        - Any IT function can be changed into a service
- Cloud service providers
    - Provide cloud services
        - SaaS, PaaS, IaaS, etc
    - Charge a flat fee or based on use
        - More data, more cost
    - You still manage your processes
        - Internal staff
        - Dev team

        - Operational support
- Managed service providers
    - Managed Service Provider (MSP)
        - Also a cloud service provider
        - Not all cloud service providers are MSP's
    - MSP support
        - Network connectivity management
        - Backups and disaster recovery
        - Growth management and planning
    - Managed Security Service Provider (MSSP)
        - Firewall management
        - Patch management, security audits
        - Emergency Response
- On-premises vs Off premises
    - On premises
        - Your application are on local hardware
        - Your servers are in your data center in your building
    - Off premises
        - Your servers are not in your building
        - They may not even be running on your hardware
        - Usually a specialized computing environment
- Cloud deployment models
    - Public
        - Available to everyone over the Internet
    - Community
        - Several organizations share the same resources
    - Private
        - Your own virtualized local data center
    - Hybrid
        - A mix of public and private

- Cloud Computing
    - Computing on demand
        - Instantly available computing power
        - Massive data storage capacity
    - Fast implementation
        - IT teams can adjust rapidly to change
        - Smaller startups cost and pay-as-you-go
    - Not always the best solution
        - Latency - the cloud is far away
        - Limited bandwidth
        - Difficult to protect data
        - Requires Internet/Network connectivity
- Edge computing

    - Over 30 billion IoT devices on the Internet
        - Devices with very specific functions
        - A huge amount of data\
    - Edge computing - “edge”
        - Process application data on an edge server
        - Close to the user
    - Often process data on the device itself
        - No latency, no network requirement
        - Increased speed and performance
        - Process where the data is, instead of processing in the cloud
- Fog computing
    - Fog
        - A cloud thats close to your data
        - Cloud + Internet of things
    - Fog computing
        - A distributed cloud architecture
        - Extends the cloud
    - Distribute the data and processing
        - Immediate data stays local - No Latency
        - Local decisions made from local data - no bandwidth requirements
        - Private data never leaves - Minimizes security concerns
        - Long term analysis can occur in the cloud - Internet only when required

---

- Designing the Cloud
    - On demand computing power
        - Click a button
    - Elasticity
        - Scale up or down as needed
    - Applications also scale
        - Access from anywhere
    - How does it happen?
        - Planning and technology
- Thin client
    - Basic application usage
        - Applications actually run on a remote server
        - Virtual Desktop Infrastructure (VDI), Desktop as a Service (DaaS)
        - Local device is a keyboard, mouse, and screen
    - Minimal operating system on the client
        - No huge memory or CPU needs
    - Network connectivity
        - Big network requirement
        - Everything happens across the wire
- Virtualization
    - Virtualization

        - Run many different operating systems on the same hardware
    - Each application instance has its own operating system
        - Adds overhead and complexity
        - Virtualization is relatively expensive
- Application containerization
    - Container
        - Contains everything you need to run an application
        - Code and dependencies
        - A standardized unit of software
    - An isolated process in a sandbox
        - Self contained
        - Apps can't interact with each other
    - Container image
        - A standard for portability
        - Lightweight, uses the host kernel
        - Secure separation between the applications
- Microservices and API's
    - Monolithic applications
        - One big application that does everything
    - Application contains all decision making processes
        - User interface
        - Business logic
        - Data input and output
    - Code challenges
        - Large codebase
        - Change control challenges
    - API's
        - Application Programming Interfaces
    - API is the glue for the microservices
        - Work together to act as the application
    - Scalable
        - Scale just the microservices you need
    - Resilient
        - Outages ate contained
    - Security and compliance
        - Containment is built in\
- Serverless architecture
    - Function as a Service (FaaS0
        - Applications are separated into individual, autonomous functions
        - Remove the operating system from the equation
    - Developer still creates the server side logic
        - Runs in a stateless compute container
    - May be event triggered and ephemeral (temporary)
        - May only run for one event

    - Managed by a third-party
        - All OS security concerns are at the third party
- Transit Gateway
    - Virtual Private Cloud (VPC)
        - A pool of resources created in a public cloud
    - Common to create many VPC's
        - Many different application clouds
    - Connect VPC's with a transit gateway
        - And users to VPC's
        - A "cloud router"
    - Now make it secure
        - VPC's are commonly on different IP subnets
        - Connecting to the cloud is often through a vpn
- Resource policies
    - Assigning permissions to cloud resources
        - Not the easiest tak
        - Everything is in constant motion
    - Specify which resources can be provisioned (Azure)
        - Create a service in a specific region, deny all others
    - Specify the resource and what actions are permitted (amazon)
        - Allow access to the an API gateway from an Ip address range
    - Explicitly list the users who can access the resource (amazon)
        - Userlist is associated with the resource
- Service integration
    - Service Integration and management (SIAM)
    - Many different service providers
        - The natural result of multisourcing
    - Every provider works differently
        - Different tools and processes
    - SIAM is the integration of these diverse providers
        - Provide a single business facing IT organization
    - An evolving set of processes and procedures

- Infrastructure in code
    - Describe an infrastructure
        - Define servers, network, and applications as code
    - Modify the infrastructure and create versions
        - The same way you version application code
    - Use the description (code) to build other application instances
        - Build it the same way every time based on the code
    - An important concept for cloud computing
        - Build a perfect version every time
- SDN (Software Defined Networking)
    - Networking devices have two functional planes of operation

        - Control plane, data plane
    - Directly programmable
        - Configuration is different than forwarding
    - Agile
        - Changes can be made dynamically
    - Centrally Managed
        - Global view, single pane of glass
    - Programmatically configured
        - No human intervention
    - Open standards / vendor neutral
        - A standard interface to the network
- SDV (Software Defined Visibility)
    - You must see the traffic to secure the data
        - React and respond
    - Dynamic deployments include security and network visibility devices
        - Next generation firewalls, web application firewalls, Security Info and Event management (SIEM)
    - Data is encapsulated and encrypted
        - VXLAN (Virtual Extensible LAN) and SSL/TLS
    - New tech change what you can see
        - Infrastructure as code, microservices
    - Security devices monitor app traffic
        - SDV provides visibility to traffic flows
    - Visibility expands as the application instances expand
        - Real time metrics across all traffic flows
    - Application flows can be controlled via API
        - Identify and react to threats

- VM sprawl avoidance
    - Click a button
        - You've built a server or multiple servers, networks, firewalls
    - It becomes almost too easy to build instances
        - This can get out of hand very quickly
    - The virtual machines are sprawled everywhere
        - You aren't sure which VMs are related to which applications
            - It becomes extremely difficult to deprovision
    - Formal process and detailed documentation
        - You should have information on every virtual object.
- VM Escape Protection
    - The virtual machine is self contained
        - Theres no way out
    - Virtual machine escape
        - Break out of the VM and interact with the host operating system or hardware

    - Once you escape the VM, you have great control
        - Control the host and control other guest VM's
    - This would be a huge exploit
        - Full control of the virtual world
- Escaping the VM
    - March 2017 - Pwn2Own competition
        - Hacking contest
            - You pwnit, you own it with some cash