Welcome

Moving to the Cloud

Hello

About me...

- 25+ years in the industry
- 20+ years in teaching
- Certified Cloud architect
- Passionate about learning
- Also, passionate about Reese's Cups!

Why study this subject?

- Cloud is everywhere and it’s not going away
- As with many topics in technology, there are multiple options and multiple dimensions to those options
- Building a deeper understanding of Cloud and its offerings positions you to help your customers with their Digital Transformation journey

We teach over 400 technology topics

You experience our impact on a daily basis!

My pledge to you

I will...

- Make this interactive
- Ask you questions
- Ensure everyone can speak
- Use an on-screen timer

Objectives

At the end of this course you will be able to:

- Explain the Cloud and respond to the changing technical environment
- Define your role in your organization’s migration to the Cloud
- Participate in migration planning
- Configure and access a simple instance of AWS infrastructure

Agenda

- Why move to the Cloud?
- Dealing with the Cloud paradigm shift
- Moving to the Cloud
- Introduction to AWS
- Working with AWS

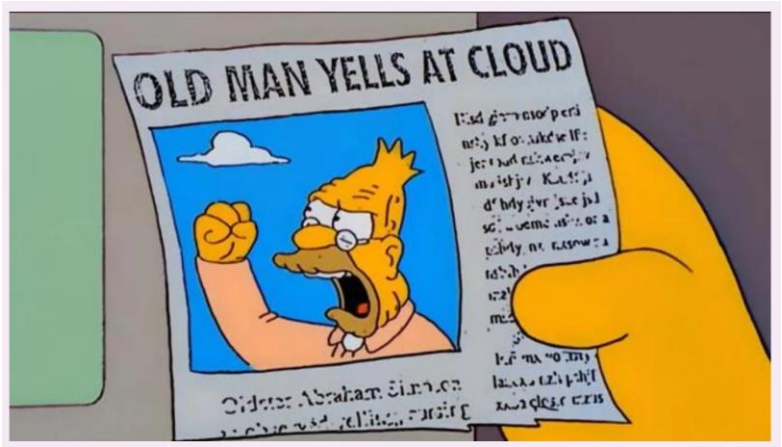
How we’re going to work together

- Lecture & slides
- Class discussions
- Demos
- Hands-on labs

Open Discussion

What is “the Cloud”?

How does “the Cloud” impact (or even enhance) our approach to business & technology enablement?



Infrastructure Options

What Are the Options?

- On-Premise
- Public Cloud
- At the Edge
- Hybrid Cloud

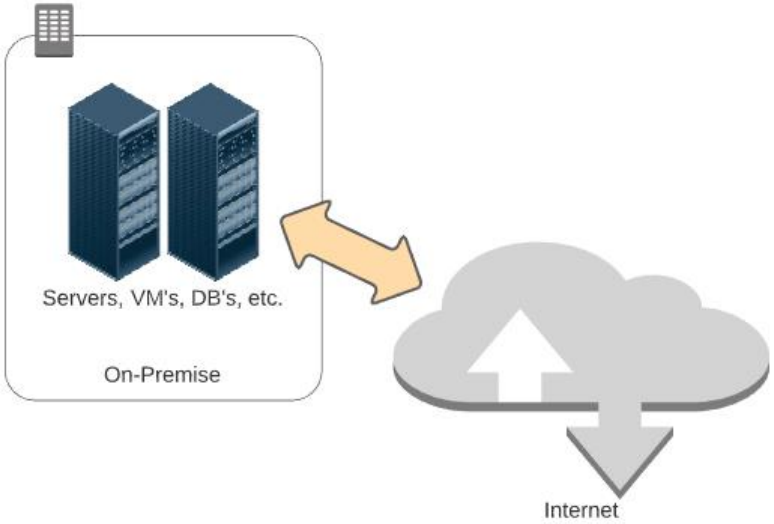
What do they all mean?

On-Premise

Can mean a few different things:

- In a wholly-owned Data Center
- In a COLO (or co-location Data Center)
- Sometimes called a “private cloud”

On-Premise



On-Premise

Why and What?

- It’s how infrastructure has traditionally been done
- With this model, companies try and estimate hardware capacity needed to support business operations
- Stakeholders look to plan out expected levels of consumption for the next 3 – 5 years (capacity to handle current volumes as well as expected growth)
- Some critical workloads may not be suitable for anything but a physical and directly-managed implementation (e.g., mainframe)

On-Premise – Discussion

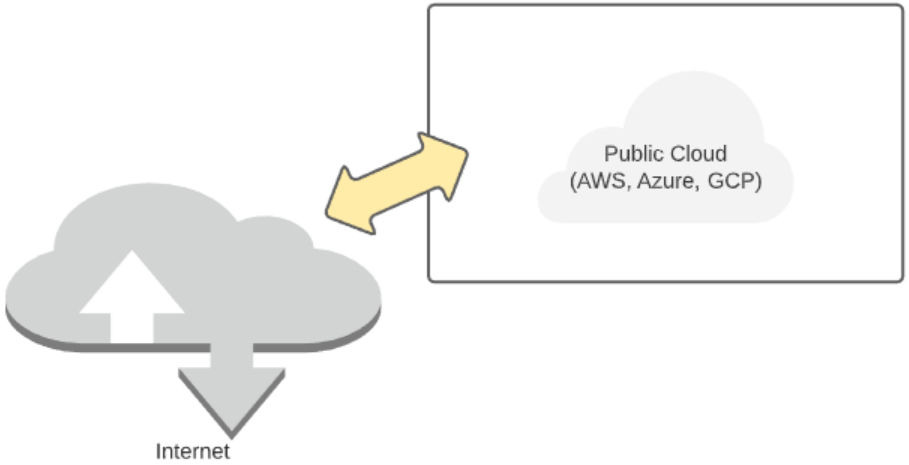
Pros?

- Discrete capacity planning (even if that planning was off)
- Some workloads (e.g., mainframes and certain legacy systems) are tailor-made for a physical data center
- With a move to COLO's, companies could begin to share expenditure

Cons?

- Sometimes difficult to know what is needed and when it is needed – if the plan was off (or unexpected spikes in demand occurred), difficult to adjust quickly
- Some workloads are just as effective (if not more so) in a virtual vs. physical implementation
- Harder to control costs and plan for costs – CAPEX vs. OPEX

Public Cloud



Public Cloud

Why and What?

- Platform using the standard “Cloud computing model” to provide infrastructure and application services
- Accessed and integrated via the Internet
- May provide a few different types of services – IaaS, PaaS, etc.
- Usually supports a subscription or “pay as you go” (on-demand) pricing model
- Largest players in this space include Azure, AWS and GCP

Public Cloud - Discussion

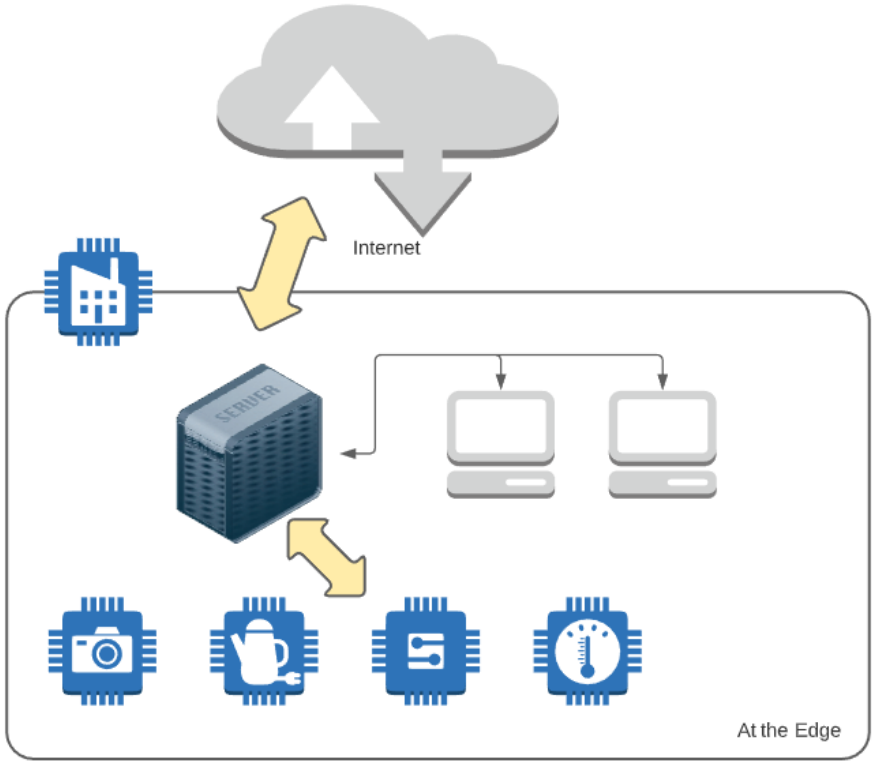
Pros?

- Flexibility and elasticity in capacity planning – enables automated schedule-based or metrics-based adjustments to capacity when required
- In some cases, managed services can be leveraged reducing operations overhead
- Because services are PAYG (pay as you go), you're only charged for what you use, and those expenses are OPEX

Cons?

- Requires enough historical data for schedule-based planning or the right configuration for metrics-based planning
- With managed services you lose some levels of granular control
- Because of the flexibility/elasticity, it can be difficult to budget and, if Cloud services are not managed/monitored, costs can be high

At the Edge

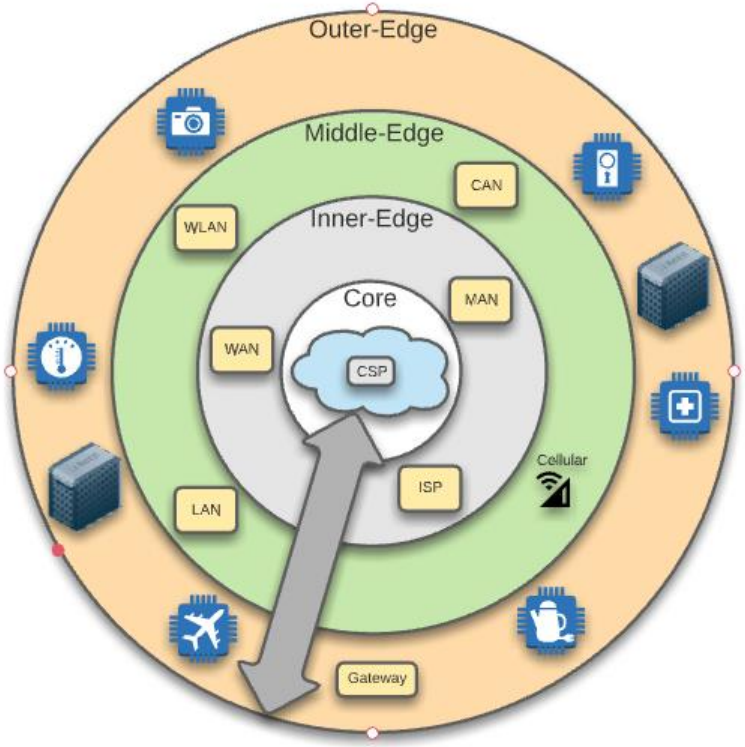


At the Edge

Can include three distinct layers:

- Inner or near edge
- Middle edge
- Outer or far edge

At the Edge – Layers



CSP – Cloud Service Provider
WAN – Wide Area Network
ISP – Internet Service Provider
MAN – Metropolitan Area Network
LAN – Local Area Network
WLAN – Wireless Local Area Network
CAN – Campus Area Network

At the Edge

Why and What?

- It’s about bringing the power of Cloud computing to you
- Enables additional processing closer to the sources of data while still supporting the offload of higher order processing to the Cloud
- Often involves setting up “Cloud-in-a-box” facilities on-premise
- IoT (Internet of Things) is a good example – devices in a facility reading massive amounts of data can incorporate processing at the edge to improve overall efficiency
- Helps inject lower latency, increased security and improved bandwidth into systems used to aggregate critical data for an enterprise

At the Edge - Discussion

Pros?

- Allows distribution of processing power across a larger surface area
- Can be used to bring critical latency, security and bandwidth improvements to specific types of business workflows
- Efficiencies gained “at the edge” can help with managing the cost of processing data

Cons?

- Requires more infrastructure and more configuration to support that distribution
- Increased distribution of processing power and activity can expand attack surface and requires the right configuration to ensure optimal interaction between system components (i.e., increased complexity)
- More components “at the edge” can lead to increased infrastructure costs

Hybrid Cloud

Why and What?

- In many ways, an amalgamation of the other options
- Supports distribution of system processing across on-premise infrastructure and the public Cloud
- Allows an enterprise to keep workloads that are best-suited for on-premise running on-premise while allowing migration of components that can move to the public Cloud
- Can help make an enterprise’s move to the Cloud more gradual and planful

Hybrid Cloud - Discussion

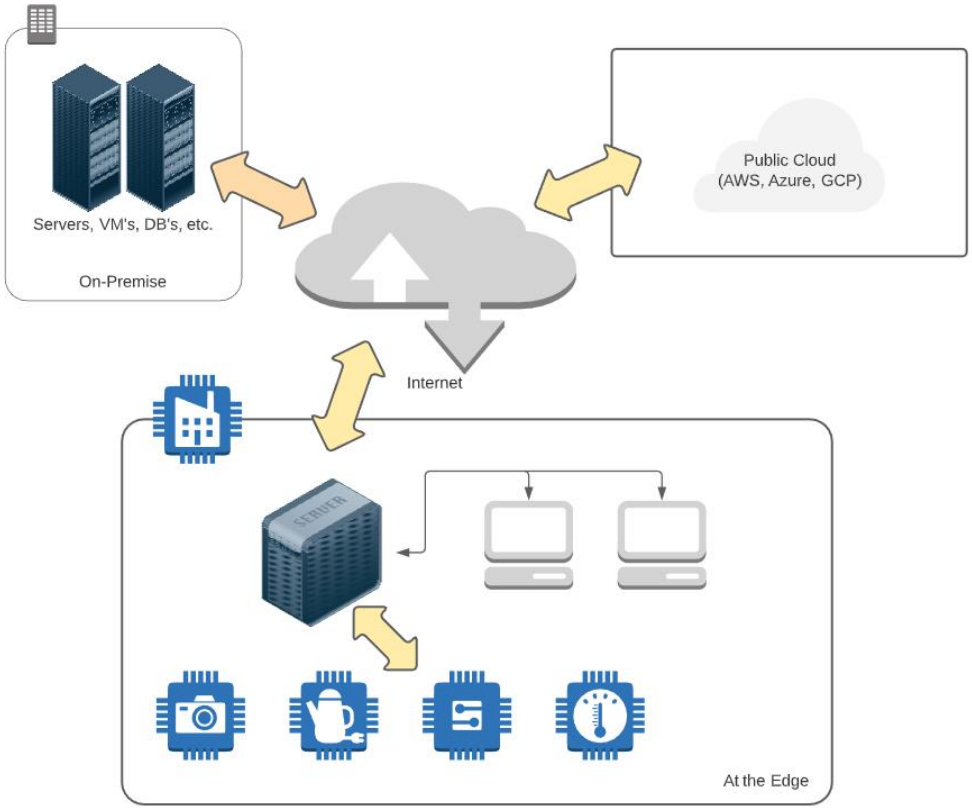
Pros?

- Allows distribution of processing power across a larger surface area
- Can allow a move to the Cloud to be more gradual and allow an enterprise to target optimal deployment platform while making the move
- The ability to support a gradual move enables an enterprise to assess and understand Cloud costs over time

Cons?

- Requires more infrastructure and more configuration to support that distribution
- As with Edge, can lead to increased complexity, often including required setup and maintenance of dedicated, secure connectivity between a data center and the Cloud
- If not managed optimally, costs can be higher due to need to pay for Cloud usage and data center (CAPEX + OPEX)

Hybrid Cloud



Lab 00 – Getting Setup

Application Hosting

What Do We Mean by Hosting?

- The target infrastructure and runtime platform that will be employed for deployment and execution of an application or system
- Can include compute (CPU and server resources), storage, network, data and operating system

Application Hosting – An “Interesting” Example?

Here’s an example of someone thinking “outside-of-the-box” when it comes to application hosting!

https://mashable.com/article/pregnancy-test-doom/

What Are the Hosting Options with Cloud?

- IaaS
- PaaS
- Serverless / FaaS
- SaaS
- Containers

What do they all mean?

Infrastructure-as-a-Service (IaaS)

- Involves the building out (and management) of virtual instances of:
  - Compute
  - Network
  - Storage

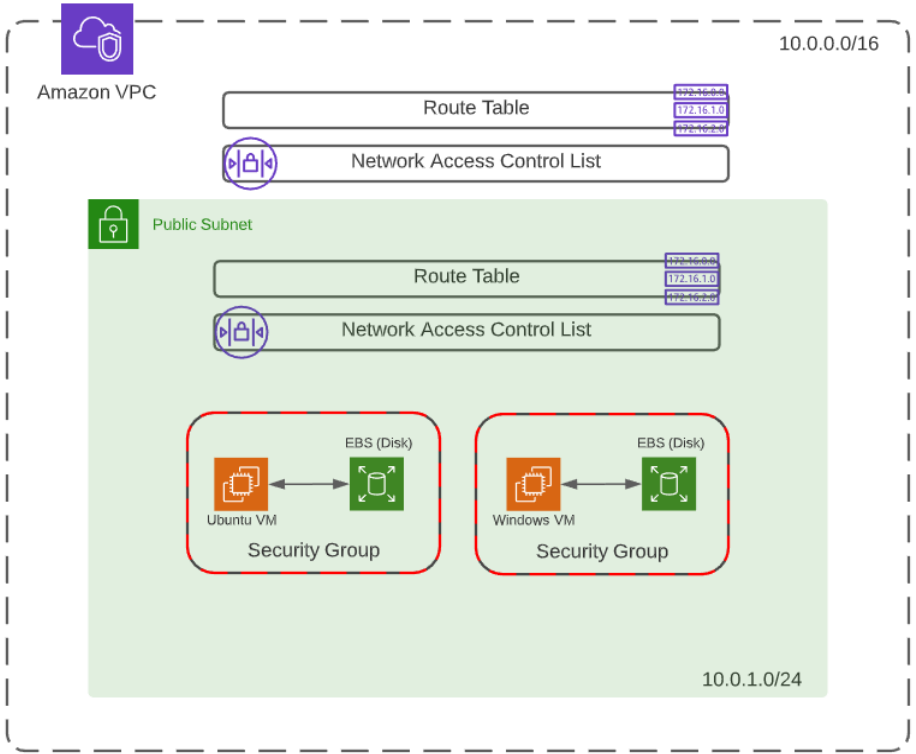
- Akin to spinning up a server (physical or virtual) in your location or data center complete with disks and required network connectivity

- The difference is in the where – instead of in your data center, it is created in a data center managed by one of the public Cloud providers

- Your organization is responsible for patching the OS, ensuring all appropriate security updates are applied and that the right controls are in place to govern interaction between this set of components and other infrastructure

IaaS – Create a VM in AWS

Demo



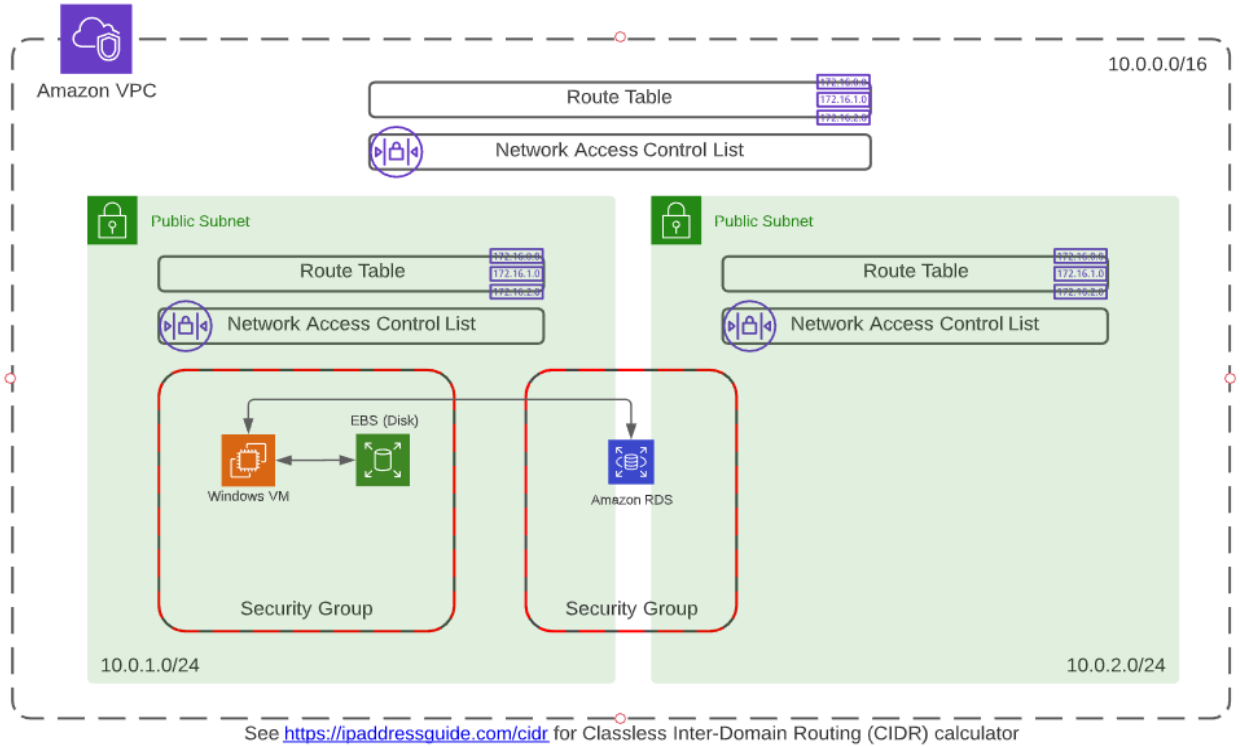
See https://ipaddressguide.com/cidr for Classless Inter-Domain Routing (CIDR) calculator

Platform-as-a-Service (PaaS)

- Involves leveraging managed services from a public Cloud provider
- With this model, an enterprise can focus on management of their application and data vs. focusing on management of the underlying infrastructure
- Patching and security of the infrastructure used to back the managed services falls to the CSP (Cloud Service Provider)
- Many managed services support automatic scale up or down depending on demand to help ensure sufficient capacity is in place
- Part of what is often termed the “Shared Responsibility Model”

PaaS – Create an RDS Instance in AWS

Demo



Serverless / Functions-as-a-Service (FaaS)

- Also represents a type of managed service provided by the CSP
- Cost structure is usually consumption-based (i.e., you only pay for what you use)
- Supports many different coding paradigms (C#/.NET, NodeJS, Python, etc.)
- Typically, with Serverless (and PaaS), the consumer is only concerned with the application code and data – elements of the CSP's "backbone" used to support are managed by the CSP
- Includes more sophisticated automated scaling capabilities – built for Internet scale

FaaS – Create a Lambda Function in AWS

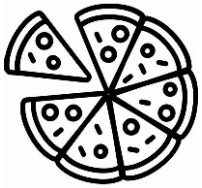
Demo

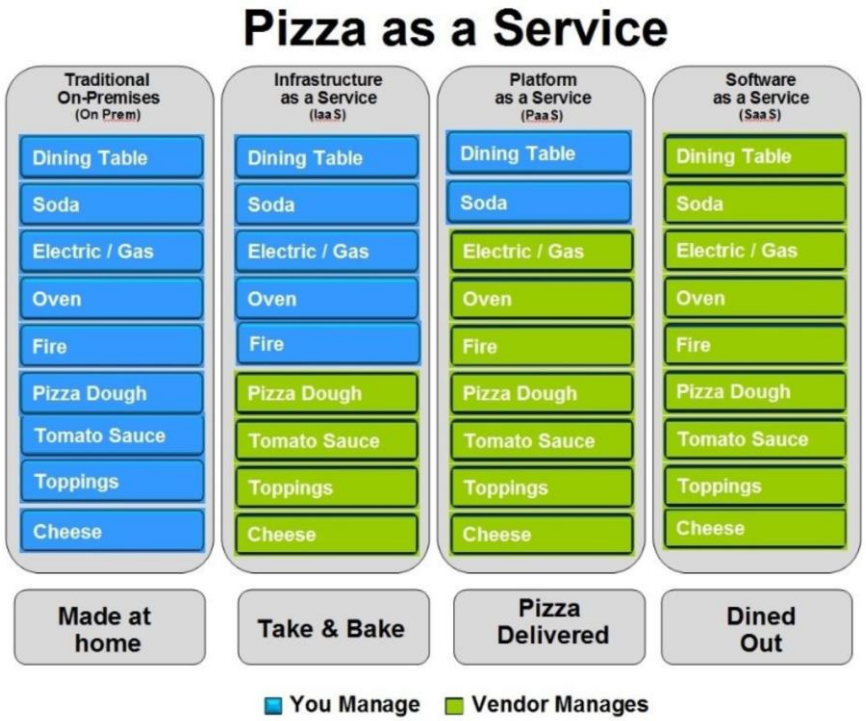
Software-as-a-Service (SaaS)

- Subscription-based application services
- Licensed for utilization over the Internet / online rather than for download and install on a server or client machine
- Fully-hosted and fully-managed by a 3rd party
- Of those discussed, often the cheapest option for service consumers
- However, also offers minimal (or no) control, outside of exposed configuration capabilities

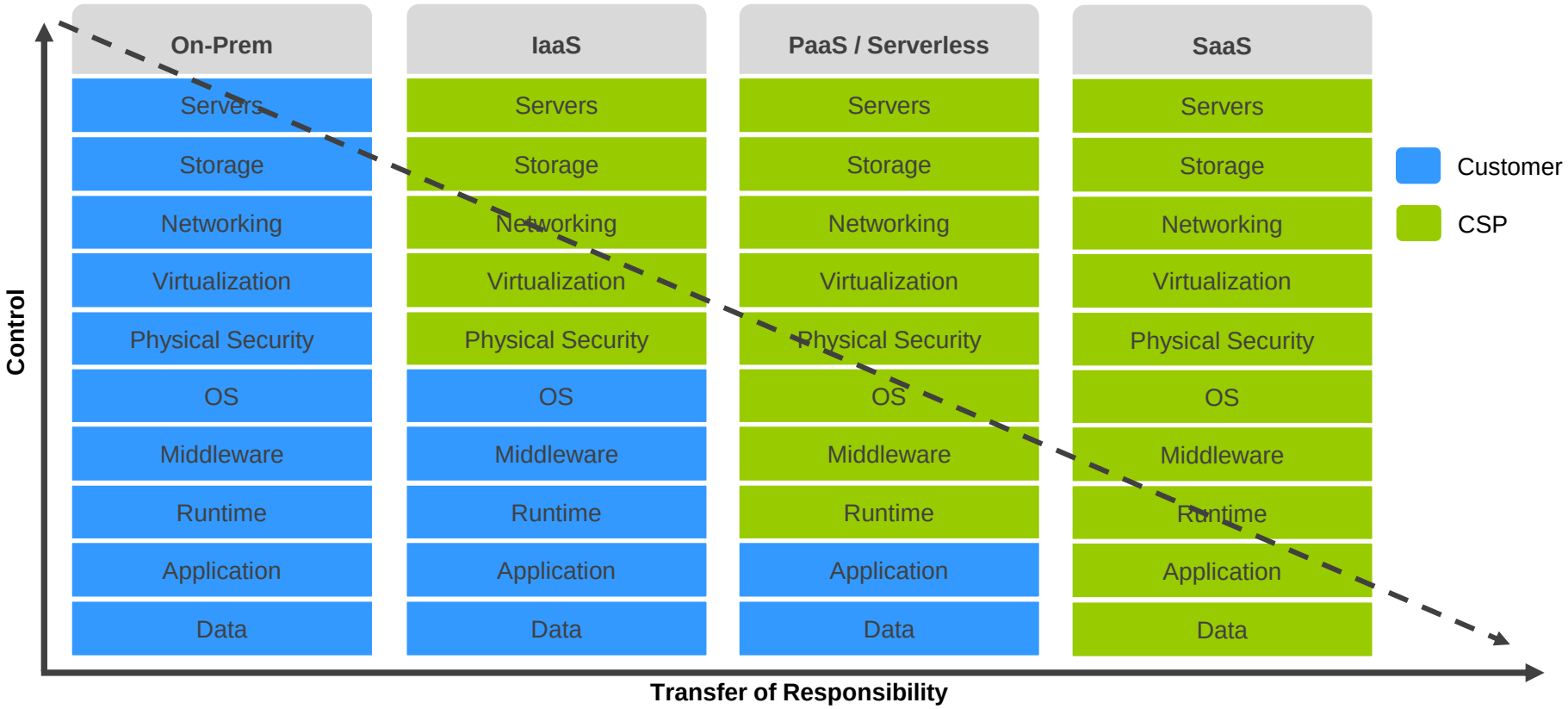
Pizza-as-a-Service

From a LinkedIn post by Albert Barron from IBM (https://www.linkedin.com/pulse/20140730172610-9679881-pizza-as-a-service/**)**





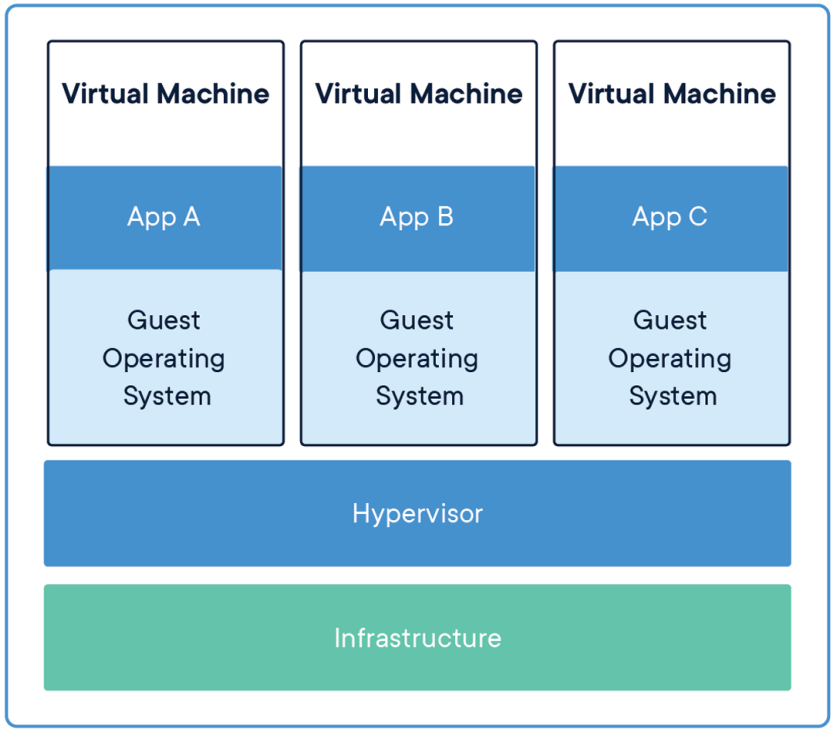
Side-by-Side Comparison



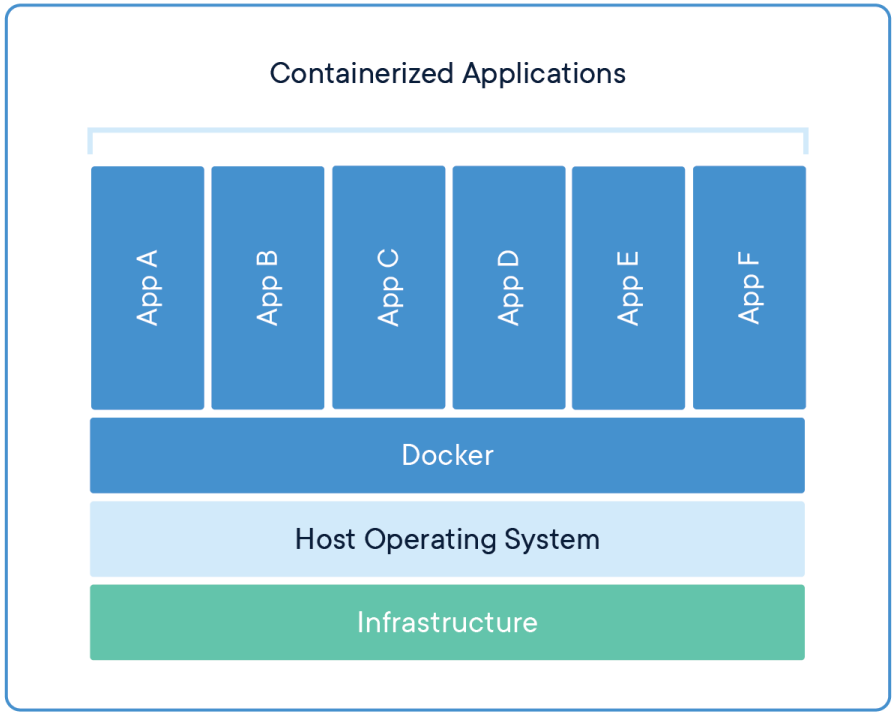
Containers

- Form of virtualization at the app packaging level (like virtual machines at the server level)
- Isolated from one another at the OS process layer (vs VM’s which are isolated at the hardware abstraction layer)
- Images represent the packaging up of an application and its dependencies as a complete, deployable unit of execution (code, runtime and configuration)

Virtual Machines



Containers


Containerized Applications
App A
App B
App C
App D
App E
App F
Docker
Host Operating System
Infrastructure

Containers

- A platform (e.g., Docker) running on a system can be used to dynamically create containers (executable instances of the app) from the defined image
- Typically, much, much smaller than a VM which makes them lightweight, quickly deployable and quick to “boot up”
- An orchestration engine (e.g., Kubernetes) might be used to coordinate multiple instances of the same container (or a “pod” of containers) to enable the servicing of more concurrent requests (scalability)

Which One is Better?

- The answer is “it depends”
- It depends on the type of application
- It depends on the enterprise
- It depends on the skillset and expertise within the organization
- It depends on whether you have budget and opportunity to modernize an application environment (in some cases)
- The best option might be a combination of multiple approaches – right tool for the right job

Case Study – Discussion

Scenario: A global supplier is looking to modernize its order processing system. Among other things, this system includes Accounts Payable, Accounts Receivable, Inventory, Shipping, Invoicing, and Raw Materials modules. Most of their current system is built on a mainframe stack which is several versions behind in terms of system upgrades and patches, and, while the modules used by the company are provided with out-of-the-box implementations by the mainframe vendor, the company has chosen to heavily customize those modules over multiple years. Internal users interface with the modules of the system using "green screen" terminals. There is a thin web layer for presentation to and utilization by external users which submits Message Queueing (MQ) calls to the mainframe for the back-end business logic.

Let's discuss:

- What kinds of requirements gathering questions would you ask this customer as you begin helping them plan for modernization?
- What kinds of infrastructure and hosting recommendations would you provide to this customer for technical implementation of a modernized solution in the Cloud?

Architecting for the Cloud

Architecting for the Future

- When we architect and build an application at a “point in time”, we hope that the application will continue to be utilized to provide the value for which it was originally built
- Since the “only constant is change” (a quote attributed to Heraclitus of Ephesus), we have to expect that the environment in which our application “lives and works” will be dynamic
- Change can come in the form of business change (change to business process), the need to accommodate innovation and ongoing advancement in technology
- Often, the speed at which we can respond to these changes is the difference between success and failure

Architecting for the Future

In order to ensure that we can respond to change “at the speed of business”, we need to build our systems according to best practices and good design principles:

- Business-aligned design
- Separation of concerns
- Loose coupling
- Designing for testability

Business-Aligned Design

- Build systems that use models and constructs that mirror the business entities and processes that the system is intended to serve
- Drive the design of the system and the language used to describe the system based around the business process not the technology
- AKA Domain Driven Design
- Results in a system built out of the coordination and interaction of key elements of the business process – helps to ensure that the system correlates to business value
- Also helps business and technology stakeholders keep the business problem at the forefront

Separation of Concerns

- Break a large, complex problem up into smaller pieces
- Drive out overlap between those pieces (modules) to keep them focused on a specific part of the business problem and minimize the repeat of logic
- Logic that is repeated, and that might change, will have to be changed in multiple places (error prone)
- Promotes high cohesion and low coupling (which we will talk about in a minute)
- Solving the problem becomes an exercise in “wiring up” the modules for end-to-end functionality and leaves you with a set of potentially reusable libraries

Loose Coupling

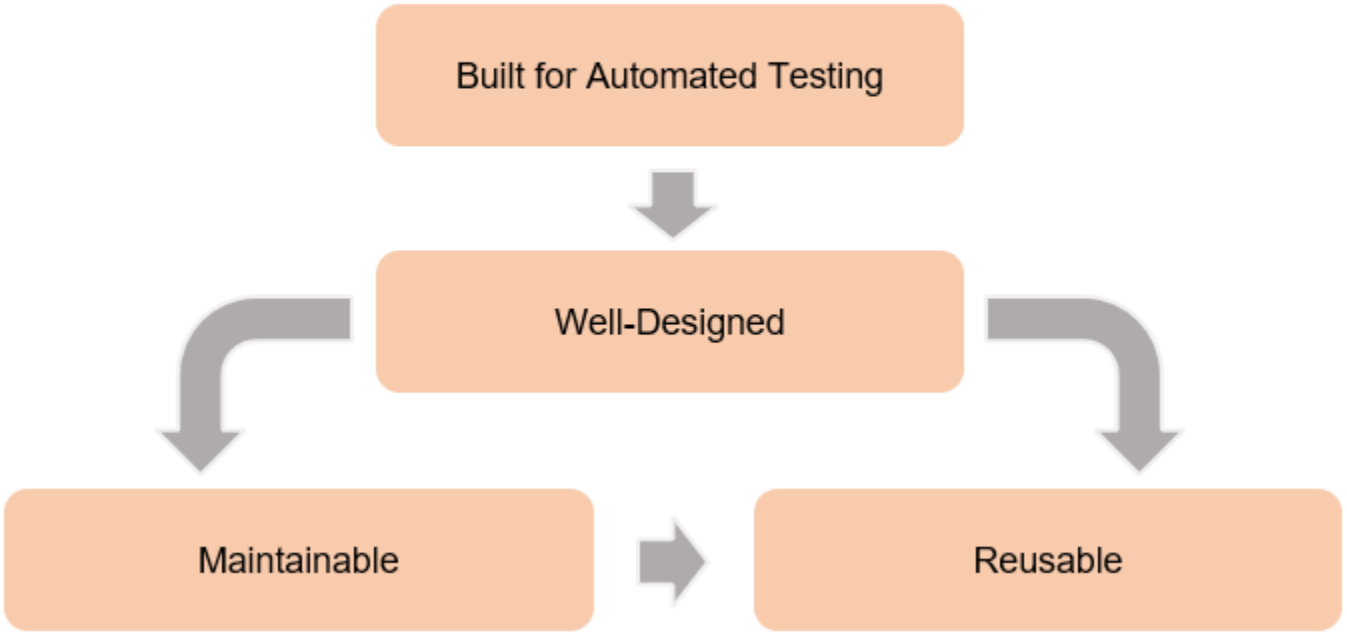
- Coupling between components or modules in a system causes problems, especially for maintaining the system over time
- System components that are tightly coupled, are more difficult to change (or enhance) – changes to one part of the system may break one or more other areas
- With coupling, you now must manage the connected components as a unit instead of having the option to manage the components in different ways (e.g. production scalability)
- Makes the job of unit testing the components more difficult because tests must now account for a broader set of logic and dependencies (e.g. tight coupling to a database makes it difficult to mock)

Designing for Testability

- Practicing the previous principles helps lead to a system that is testable
- Testability is important because it is a key enabler for verifying the quality of the system – at multiple stages along the Software Development Lifecycle (SDLC)
- When building a system, quality issues become more expensive to correct the later they are discovered in the development lifecycle – good architecture practices help you test early and often
- Ideally, testing at each stage will be automated as much as possible in support of quickly running the tests as and when needed

Designing for Testability

Testable code is…



Monolith vs. Microservices

What are Microservices?

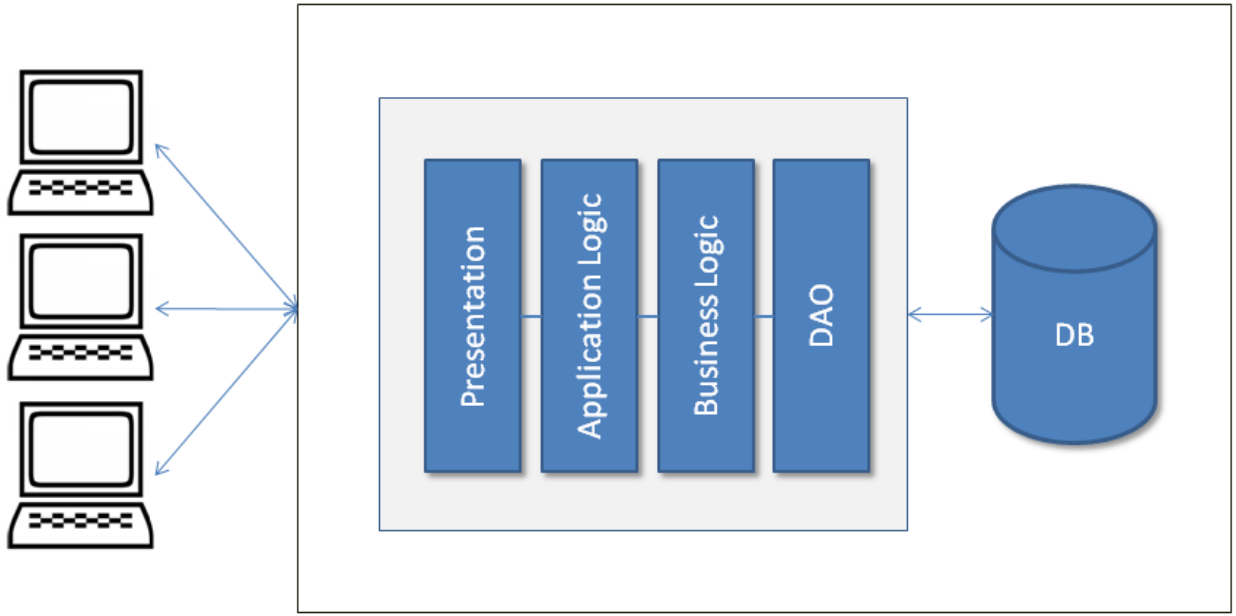
- An architectural style in which a distributed application is created as a collection of services that are:
  - Highly maintainable and testable
  - Loosely coupled
  - Independently deployable (by extension, independently-scalable)
  - Domain centric and organized around business capabilities

- The Microservices architecture enables the continuous deployment and delivery of large, complex distributed applications

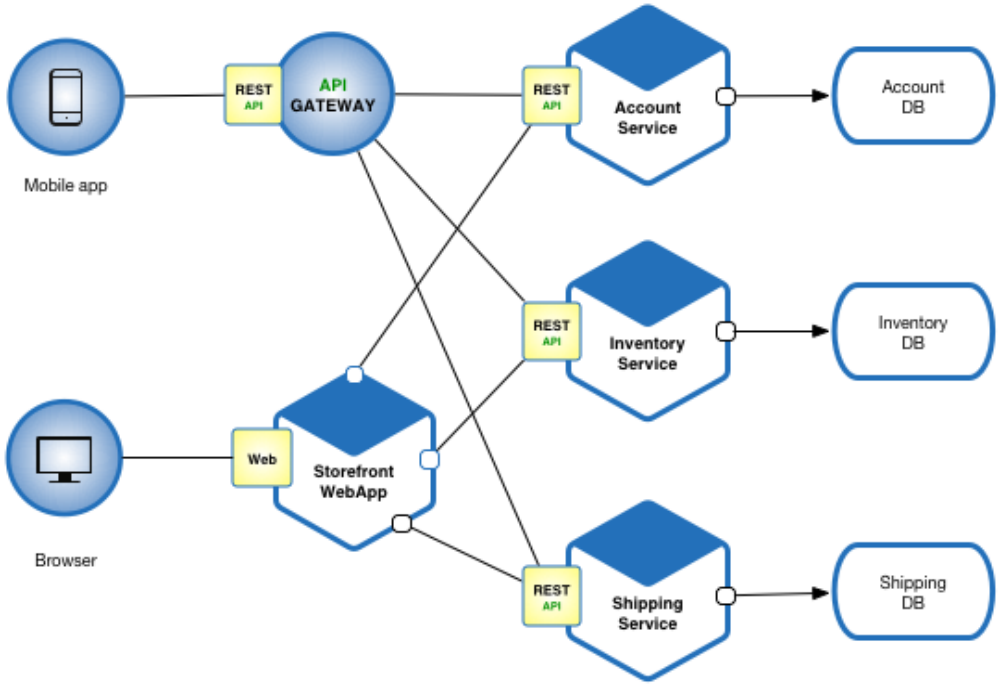
What is a Monolith?

- Typical enterprise application
- Large codebases
- Set technology stack
- Highly coupled elements
- Whole system affected by failures
- Scaling requires the duplication of the entire app
- Minor changes often require full rebuild

Monolithic Architecture Example



Example Microservices Architecture



Microservices – Benefits vs. Costs

Benefits:

- Enables work in parallel
- Promotes organization according to business domain
- Advantages from isolation
- Flexible in terms of deployment and scale
- Flexible in terms of technology

Microservices – Benefits vs. Costs

Costs:

- Requires a different way of thinking
- Complexity moves to the integration layer
- Organization needs to be able to support re-org according to business domain (instead of technology domain)
- With an increased reliance on the network, you may encounter latency and failures at the network layer
- Transactions must be handled differently (across service boundaries)

Microservices & Good Architecture

- With microservices, key concepts:
  - Cohesion (goal to increase)
  - Coupling (goal to decrease)

- SOLID principles & DDD (Domain Driven Design) lay the foundation for:
  - Clean, logical organization of components
  - Maintainability
  - Clear boundaries, encapsulation and separation of concerns in the components used to build out complex systems
  - Techniques that minimize coupling
  - Being “surgical” with our change

So, What Does This Have to Do With the Cloud?

- Complex Cloud-enabled workflows are composed of multiple systems
- Those multiple systems are composed of multiple components
- Those multiple components are composed of multiple modules or classes

So, What Does This Have to Do With the Cloud?

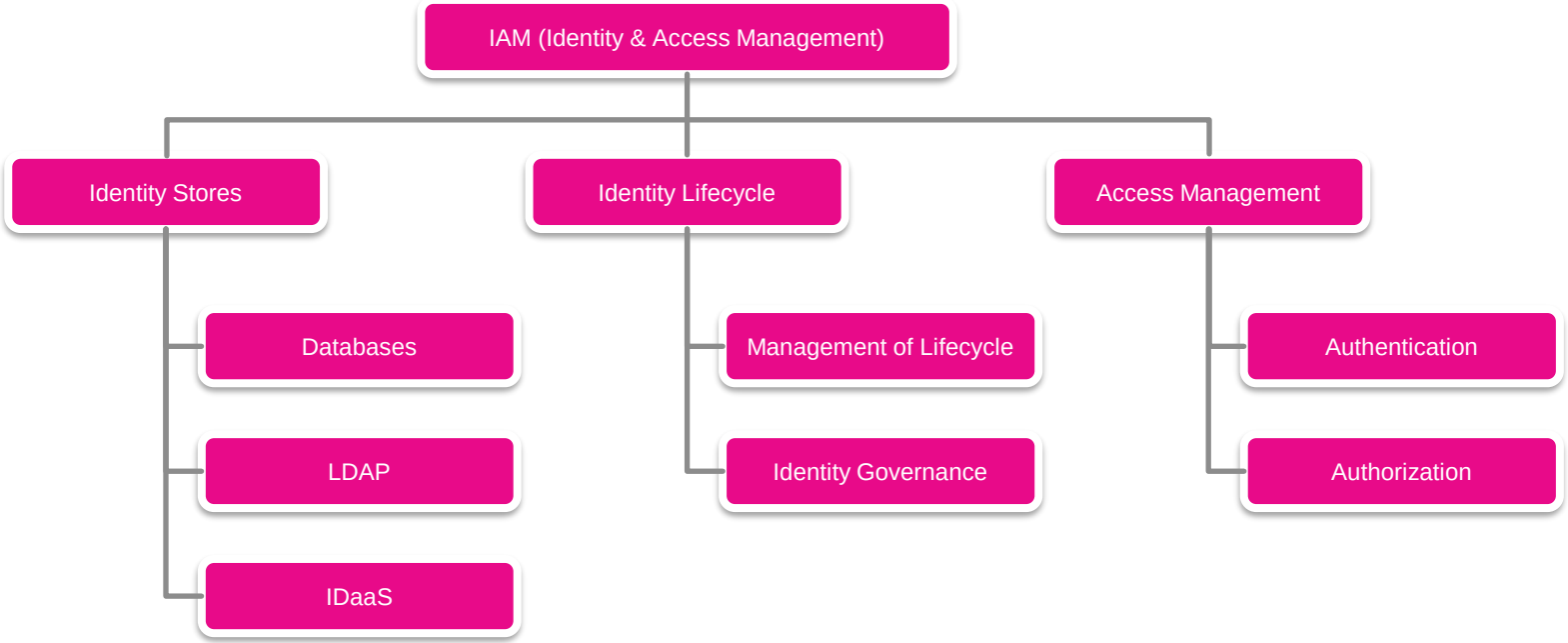
- Those multiple modules are composed of multiple routines or blocks of code
- To help ensure quality, testability, stability and maintainability of the complex Cloud-enabled workflows, we must employ good design and build practice in each building block (like a fractal)
- Furthermore, this requires forethought and intentionality – it doesn’t happen by accident

Well Architected Framework – AWS

https://aws.amazon.com/blogs/apn/the-5-pillars-of-the-aws-well-architected-framework/

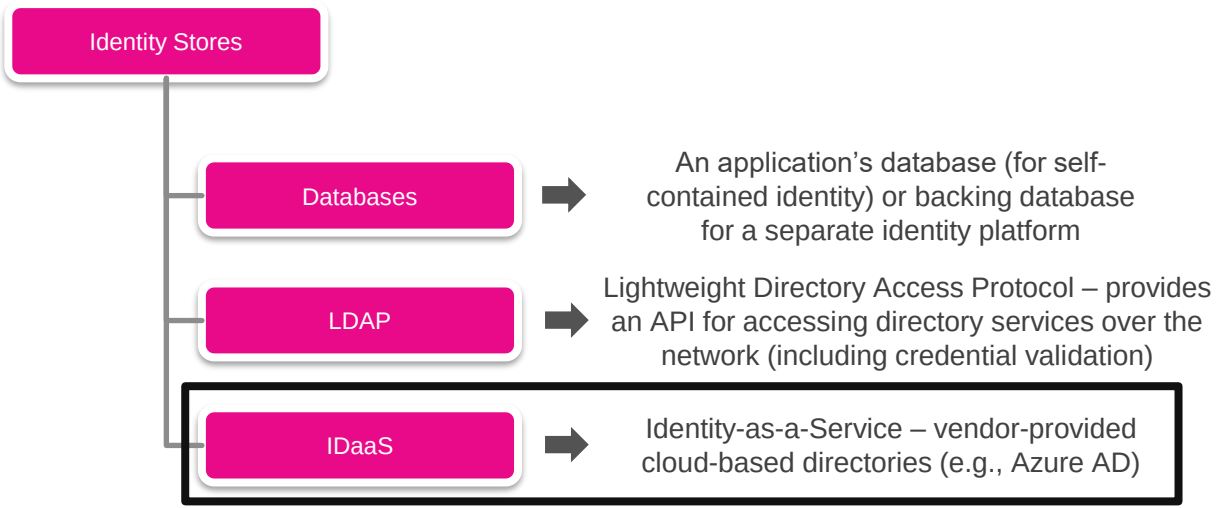
Identity Providers & AuthN/AuthZ

Identity & Access Management (IAM)



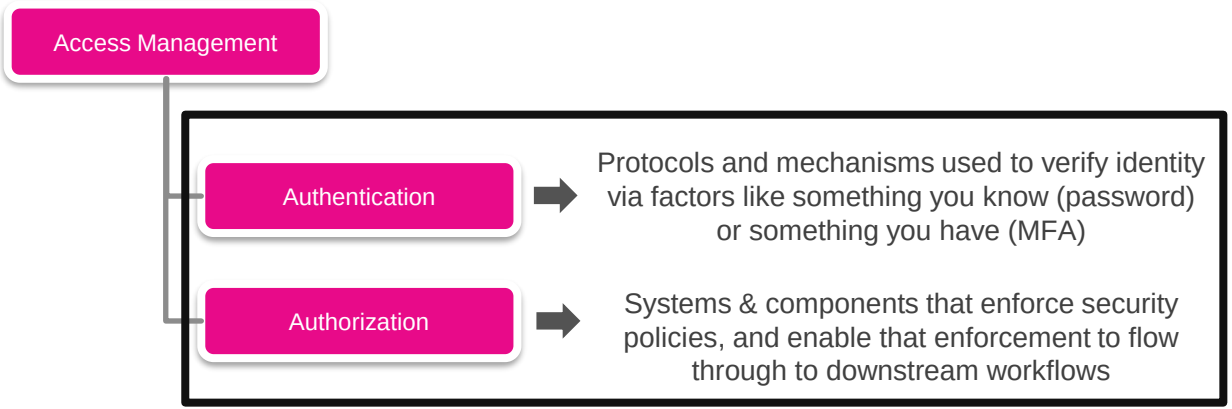
Identity Stores

- Authoritative system(s) of record for identities and their metadata
- Likely include user and non-user identities

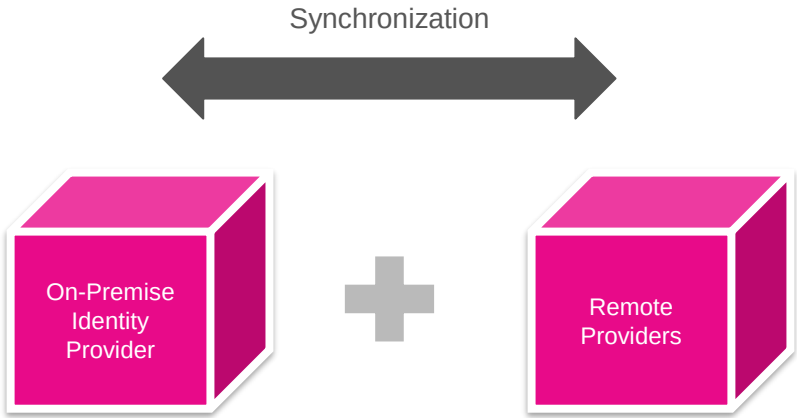


Access Management

- Verifying that a user is who they say they are
- Assessing & validating what a verified user is allowed to do & access



Securely Integrating Identity Providers



- For example, Active Directory or an on-premise installation of a platform like Okta for enabling SSO
- Often found in organizations with a traditional Data Center approach to infrastructure

- Platforms like AWS IAM, Azure Active Directory (AAD), or Okta Cloud
- Support modern AuthN/AuthZ mechanisms like OIDC and OAuth2
- Often include connectors to enable synchronization between an on-premise provider and a remote provider

Why Important for Cloud?

- Security is ALWAYS important, regardless of how & where hosted
- Exposing services & resources to the Internet invites additional layers of risk
- With less direct control over your infrastructure, applications, & data (depending on how hosted), verifying the veracity of the identities utilizing is essential
- Also, tightly controlling what those identities are allowed to do is a must to minimize exposure
- Balancing the advantages of global accessibility against critical controls

Case Study – Discussion

Scenario: A global supplier is looking to modernize its order processing system. Among other things, this system includes Accounts Payable, Accounts Receivable, Inventory, Shipping, Invoicing, and Raw Materials modules. Most of their current system is built on a mainframe stack which is several versions behind in terms of system upgrades and patches, and, while the modules used by the company are provided with out-of-the-box implementations by the mainframe vendor, the company has chosen to heavily customize those modules over multiple years. Internal users interface with the modules of the system using "green screen" terminals. There is a thin web layer for presentation to and utilization by external users which submits Message Queueing (MQ) calls to the mainframe for the back-end business logic.

Let's discuss:

- For the case study we've been covering, where is Authentication required?
- Where is Authorization required?
- What might a Cloud-enabled solution for identity & access management look like in this scenario?

Migration vs. Modernization

Application Portfolio Assessment

- Helps with planning for the migration of an enterprise’s system and software estate to the Cloud
- During this process, applications created or utilized by the enterprise will be reviewed for migration disposition
- Assessment may be completed for a small subset of the apps (as a starting point or proof of concept) or may include the entire estate

Application Portfolio Assessment

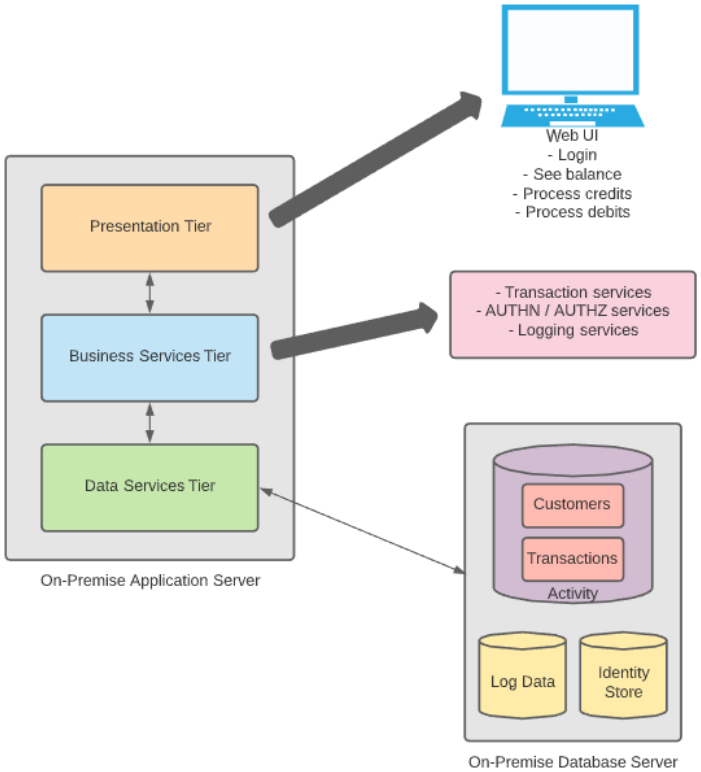
- Through this analysis, a disposition for Cloud readiness will be assigned to each app (based on several factors which we'll review shortly)
- There may also be a determination of which apps are best suited to move first, on the way to defining a plan for migration of all apps in waves
- The assessment will likely include an interview with the application owner (to gain SME knowledge about the application)

Application Portfolio Assessment

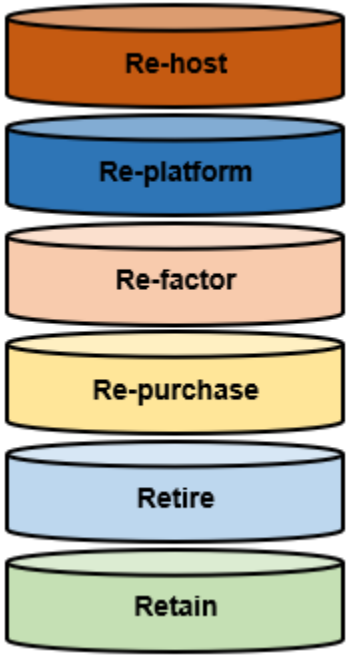
- Will likely include
  - Review of the current state architecture
  - Discussion with stakeholders about roadmap and plans for target state
  - Discussion on expected timing

- The goal is to ensure that migration efforts are focused on the activities that will bring greatest business benefit against optimized cost

Banking App – Current State



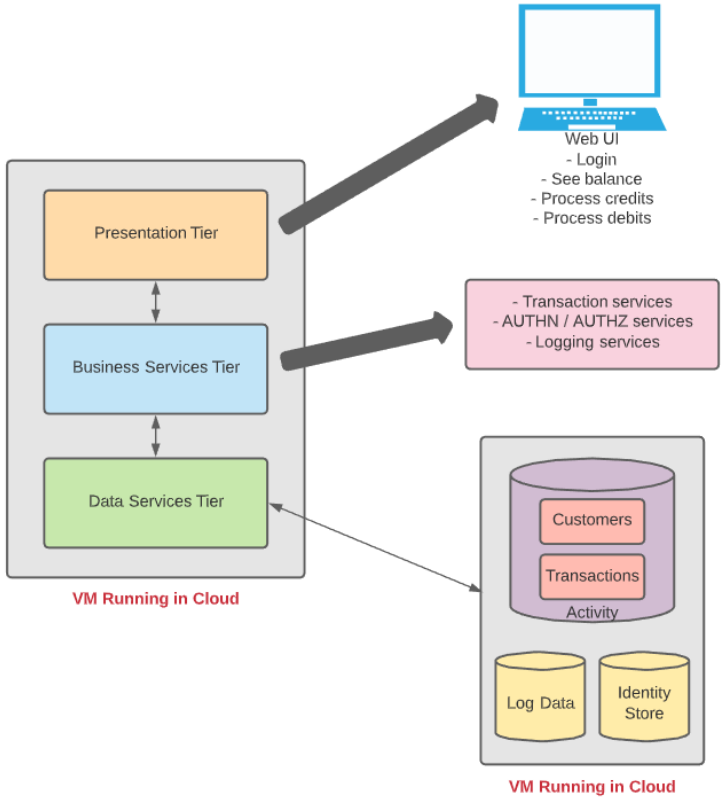
The 6 R's – Application Migration Strategies



- AKA “lift & shift”
- For all intents and purposes, involves recreating the on-premise infrastructure in the Cloud
- Sometimes used to expedite retirement of a data center

- Can be a mechanism to quickly migrate workloads and see immediate cost savings, even without Cloud optimizations
- There are third-party tools available to help automate the migration
- Once the application is in the Cloud, it can be easier to apply Cloud optimizations vs. trying to migrate and optimize at the same time

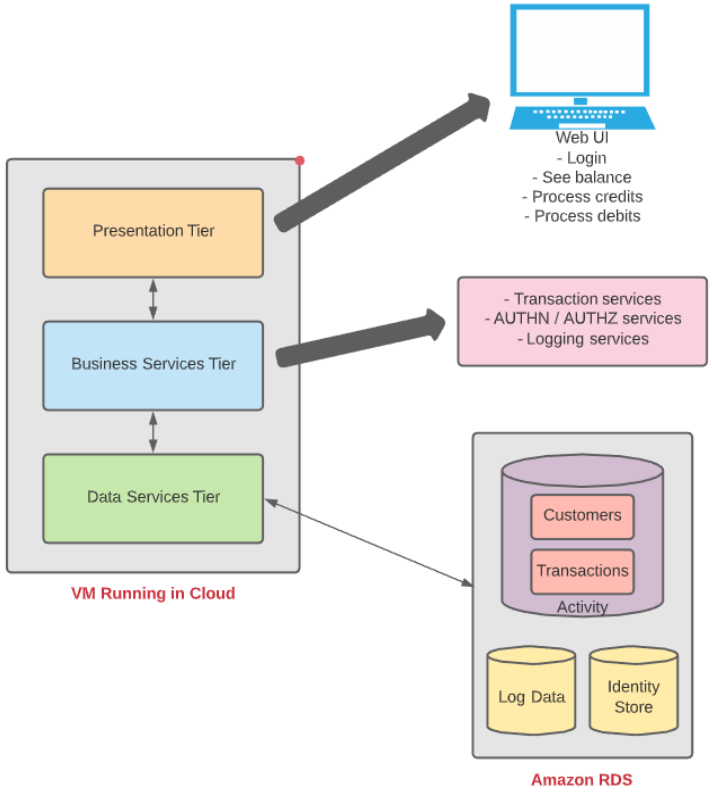

Re-host

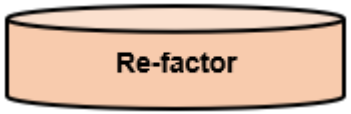

Web UI
- Login
- See balance
- Process credits
- Process debits
Presentation Tier
Business Services Tier
Data Services Tier
- Transaction services
- AUTHN / AUTHZ services
- Logging services
VM Running in Cloud
Customers
Transactions
Activity
Log Data
Identity Store
VM Running in Cloud

- AKA “lift, tinker & shift” or “lift & fiddle”
- Core architecture of the application will not change
- Involves recreating most of the on-premise infrastructure in the Cloud with a few Cloud optimizations

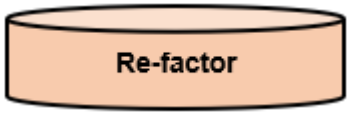
- Those optimizations will involve a move to one or more Cloud native services for a specific, tangible business benefit
- A common example is moving to a managed database (e.g. Relational Database Service, or RDS, in AWS)
- Enables migration speed while providing cost savings or benefit in a targeted portion of the application’s architecture


Re-platform

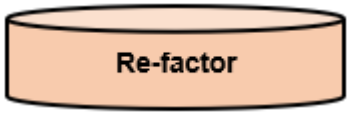

Web UI
- Login
- See balance
- Process credits
- Process debits
Presentation Tier
Business Services Tier
Data Services Tier
VM Running in Cloud
- Transaction services
- AUTHN / AUTHZ services
- Logging services
Customers
Transactions
Activity
Log Data
Identity Store
Amazon RDS



- Involves rearchitecting the application to maximize utilization of Cloud native optimizations
- Likely the most expensive and most complex of the available options
- The application profile needs to fit, and business value must be identified commensurate with the cost required to execute



- Good option if the application can benefit from features, scalability or performance offered by the Cloud
- Examples include rearchitecting a monolith to microservices running in the Cloud or moving an application to serverless technologies for scale



?

To Be Determined

- Good fit for applications that are candidates for moving from on-premise licensing (for installed products) to a SaaS model
- Often applications that have been installed to manage a specific type of business capability
- Examples can include Customer Relationship Management (CRM), Enterprise Resource Planning (ERP), HR or e-mail (among others)

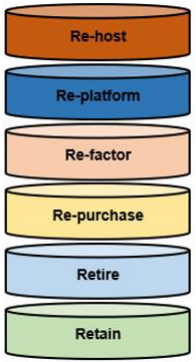
- In some cases, an enterprise may have a “healthy” percentage of legacy applications that are no longer being used or maintained (especially for larger organizations)
- When completing the Application Portfolio Assessment and associated interviews with application owners, look for opportunities to recommend retire of the unused legacy apps
- This type of application can represent a “quick win”

- The associated savings can potentially be advantageously factored into the business case for the Cloud modernization effort
- Finally, retiring unused apps reduces attack surface area from a security perspective

- Applications that must remain in place but that cannot be migrated to the Cloud without major refactor
- This type of application profile may prevent the ability to completely move out of the data center (at least in the interim)
- From a cost perspective, it can be difficult for an enterprise to take on both the operating expense of Cloud while continuing to carry the capital expense of a data center and on-premise infrastructure
- The hybrid model can be a good solution to support where needed

Case Study – Discussion

Scenario: A global supplier is looking to modernize its order processing system. Among other things, this system includes Accounts Payable, Accounts Receivable, Inventory, Shipping, Invoicing, and Raw Materials modules. Most of their current system is built on a mainframe stack which is several versions behind in terms of system upgrades and patches, and, while the modules used by the company are provided with out-of-the-box implementations by the mainframe vendor, the company has chosen to heavily customize those modules over multiple years. Internal users interface with the modules of the system using "green screen" terminals. There is a thin web layer for presentation to and utilization by external users which submits Message Queueing (MQ) calls to the mainframe for the back-end business logic.



Let's discuss:

- For the case study we've been covering, which of the 6 R's do you think we could leverage for this system?

Cloud Solution Providers (CSPs)

Cloud Solution Providers (CSPs)

- Players in the space:
  - AWS (Amazon Web Services)
  - Microsoft Azure
  - GCP (Google Cloud Platform)

AWS (Amazon Web Services)

- Management console accessible at https://aws.amazon.com
- Identity & Access Management handled via the IAM service
- Users, groups, roles, and policies
- Unit of access & activity is the “account”

AWS (Amazon Web Services)

- Uses concept of “region” for geographical location of resources
- Multiple geographies and multiple availability zones within geography

AWS (Amazon Web Services)

- See https://aws.amazon.com/about-aws/global-infrastructure/regions_az/ for additional detail

Developing for Cloud

Developing for Cloud

- Key areas of focus
  - Compute
  - Storage
  - Networking
  - Database

Compute in AWS

- Includes
  - EC2 (Elastic Cloud Compute – VM)
  - Lambda
  - Elastic Beanstalk

Storage in AWS

- Includes
  - S3 (Simple Storage Service)

Networking in AWS

- Includes
  - VPCs (Virtual Private Cloud)
  - Subnets
  - API Gateway
  - Route 53

Database in AWS

- Includes
  - RDS (Relational Database Service)
  - Amazon DynamoDB
  - Amazon Aurora

Other Available Services in AWS

- Includes
  - Machine Learning
  - IoT (Internet of Things)
  - Big Data Management
  - Many others…

Cost Management

- CAPEX vs. OPEX
- AWS → https://calculator.aws/
- Azure → https://azure.microsoft.com/en-us/pricing/calculator/
- GCP → https://cloud.google.com/products/calculator/
- Key is – ensuring accounting for ALL components
- Drives ROI (Return on Investment) & CBA (Cost-Benefit Analysis)

Case Study – Discussion

Scenario: A global supplier is looking to modernize its order processing system. Among other things, this system includes Accounts Payable, Accounts Receivable, Inventory, Shipping, Invoicing, and Raw Materials modules. Most of their current system is built on a mainframe stack which is several versions behind in terms of system upgrades and patches, and, while the modules used by the company are provided with out-of-the-box implementations by the mainframe vendor, the company has chosen to heavily customize those modules over multiple years. Internal users interface with the modules of the system using "green screen" terminals. There is a thin web layer for presentation to and utilization by external users which submits Message Queueing (MQ) calls to the mainframe for the back-end business logic.

Let's discuss:

- What types of Cloud services (either from the previously discussed list or others) would you recommend using for this modernization effort?
- How could we best manage cost estimates, capacity planning, & an ROI (Return on Investment) assessment for this use case? Where might there be "hidden" costs?

Developing on Cloud

Developing on Cloud - AWS

- Includes
  - AWS CloudFormation
  - AWS Cloud9
  - AWS Control Tower

AWS CloudFormation

- In AWS, provides either a YAML or JSON-based description language for defining infrastructure
- Supports parameterization, inputs, outputs, certain control structures
- Let’s look…

Terraform

- While Cloud components can be created manually in the Management Console, it’s not ideal
- IaC (Infrastructure-as-Code) is a better alternative
- Infrastructure code (like all other code) is code
- Can be test, versioned, put in source control, etc.
- Multiple options available for tooling to support – Terraform in addition to CloudFormation+

Operating in the Cloud

Monitoring Across Hybrid Cloud

- Monitoring & logging are key considerations in any Cloud environment
- Systems (you hope) will be running around-the-clock – maximizing business benefit
- Unless you want to directly “babysit” those systems around-the-clock, you will need automated monitoring, logging and alerting to notify you of any issues
- Allows you to optimize handling for those exceptional cases when there is a problem

Monitoring Across Hybrid Cloud

- Capabilities exist to log and aggregate log data from the public Cloud
- There are likely already processes in place to monitor on-premise systems
- The goal is a strategy that allows you to pull that data together so you can analyze and make decisions holistically

Monitoring Across Hybrid Cloud

- Key tasks include:
  - Discovery – where are the critical data sources and how do I connect
  - Aggregation – bringing the data together in a systematic way
  - Normalization – converting data from disparate data sources into a canonical format
  - Security – data scrubbing (if required) and prevention of exposure of sensitive data

- Not just about identifying problems but also using the data to effectively identify opportunities

Monitoring Across Hybrid Cloud

Potential Challenges

- You will need secure and performant connectivity between your on-premise and Cloud components
- Data formats may be very different between the different systems comprising your Hybrid Cloud environment
- You will need a strategy for gaining intelligence from the aggregated data while driving the benefit of that intelligence back into disparate systems

Elastic Scalability

- As highlighted previously, one of the main “draws” for Cloud is the ability to quickly scale up or scale down workloads

- In concert with virtualization & orchestration, the Cloud allows the automated spin up of “more” to handle:
  - Response to a specific schedule event (e.g., seasonal demand)
  - Response to an alert from a monitored event indicating that current configuration is being taxed with volume (using multiple metrics)

- It is elastic because the platform supports both scale up and down

- Key to optimizing cost vs. capability – paying for only what you need when you need it

Elastic Scalability

Potential Challenges

- Being able to determine what is needed and when can be challenging
- In the Hybrid Cloud environment, scalability may look different depending on whether you're talking on-premise services vs. Cloud services
- Determining optimal what & when may require usage data that you don't yet have with a newly deployed system
- Balancing capability against cost and ensuring "just enough"

Business Continuity/Disaster Recovery (BC/DR)

- A BC/DR strategy enables a company to plan for continued operations even in the face of a regional disaster
- Usually geographically-based – instances of services existing in *both* a primary region and in another physically-separated, secondary region
- That way, if the primary region goes down (for whatever reason), theoretically the company could continue to do business
- Doesn’t have to be a permanent issue – could be a transient failure

Business Continuity/Disaster Recovery (BC/DR)

- Design and operational considerations:
  - Latency – because of physics, data can only travel over-the-wire at a certain speed
  - Active-Active or Active-Passive – does the system require / support actively servicing requests in both geographic locations at the same time?
  - Cost – depending on the configuration, a company may be required to pay for 2x the infrastructure

Business Continuity/Disaster Recovery (BC/DR)

Potential Challenges

- As discussed, latency can be a challenge – will a secondary region perform at the level needed to meet your SLA's?
- If the profile is Active-Active, it can be challenging to coordinate data collection and intelligence gathering across the two regions
- If the profile is Active-Passive, what is the process for spinning up the secondary region, how do you keep data in sync (and then undo once the disaster scenario resolves)?
- As with elastic scalability, balancing capability against cost and ensuring "just enough"

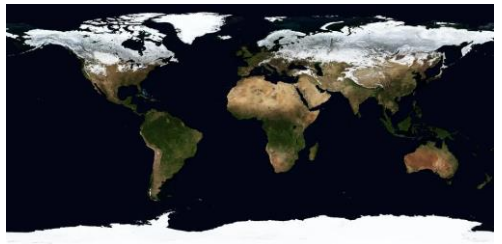
Standards & Compliance Categories

Can include:

- By geographical region
- By industry
- By technology

Standards & Compliance by Region

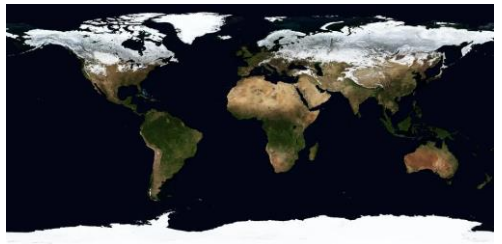
- Standards and compliance enforcement can vary by area of the world
- For example, the EU likely has different requirements than the US:
  - Federal Communications Commission (FCC) certification in the US
  - General Data Protection Regulation (GDPR) in the EU
- Other likely apply regardless of locality
  - PCI
  - SOX



Standards & Compliance by Region

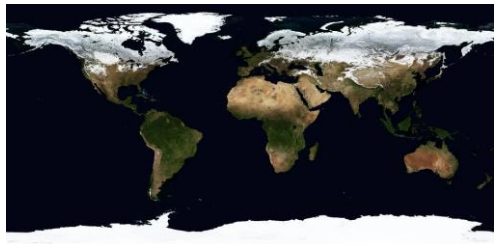
Can include considerations for:

- How data is transmitted
- How data is secured, managed, and used
- Physical or systems security of the device or Edge component itself



Standards & Compliance by Region

- Failure to adhere can limit ability to do business in the region
- Or can result in significant penalties and/or reputational damage
- Can add permutations to approach to build out of the tech



Standards & Compliance by Industry

- Different industries may have different regulations
- There can also be a difference in physical requirements
- Think remote oil field vs. data center vs. nuclear power plant

Standards & Compliance by Industry

- Regulations often driven by types of data being gathered
- Medical devices likely subject to HIPAA regulations
- Point-of-Sale (POS) devices may require PCI compliance

Standards & Compliance by Industry

- Depending on the industry, failure to comply may have devastating impact
- Think potential exposure for autonomous vehicles, for example

Standards & Compliance by Technology

- Azure Security → https://invidgroup.com/how-secure-is-microsoft-azure/
- AWS Security → https://docs.aws.amazon.com/whitepapers/latest/aws-overview/security-and-compliance.html
- GCP Security → https://medium.com/google-cloud/is-my-data-safe-in-cloud-41608c1d1f89

Assessing Security Risks

- To secure a solution, attack surfaces and potential threats must be identified
- Common practice utilizes something called threat modeling
- Includes modeling and analyzing possible attack vectors based on application

Assessing Security Risks

- Risk assessment should account for different “zones” of execution
- Security requirements for device in remote oil field different from secure data center
- And, ideally, threat modeling would be executed during design & dev phases

Threat Modeling

Securing Data in Motion

- Security required as data flows through the ether between producer and consumer
- If attacker able to intercept information flowing between the two:
  - Potentially exposes sensitive information contained within header or payload
  - Could allow insertion of alternate, damaging detail or control instruction

Securing Data in Motion

- Certificate/secrets-based Transport Layer Security (TLS) can be used to protect
- Highlights need to protect security keys
- Impact can range from trivial to devastating (depending on application)

Securing Data at Rest

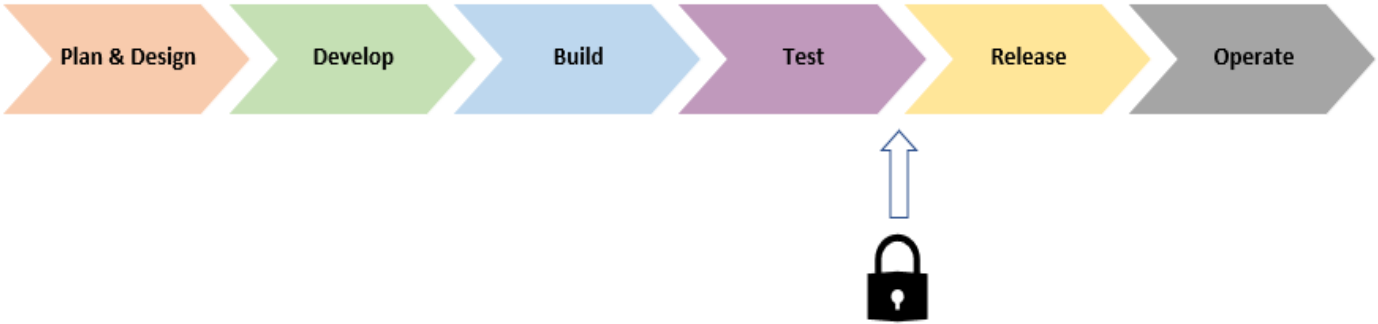
- Aggregated data stored in plain text can create a vulnerability
- In previous topic on data management, goal is gained intelligence from the data
- If the data at rest has been compromised:
  - May lead to inaccurate conclusions from analysis
  - Could provide competitor or bad actor access to a company's competitive advantage
- As with "in motion", certificate-based encryption in storage is key

DevOps & Automation

- Presents opportunities to apply scripting to:
  - Automate onboarding, offboarding, and configuration
  - Deployment & configuration of Edge components
  - Deployment & configuration of Cloud services used to aggregate & analyze data
- Practicing principles of DevSecOps helps ensure security is “shifted left”

DevOps & Automation

What is often done:



What are the potential challenges with taking this approach?

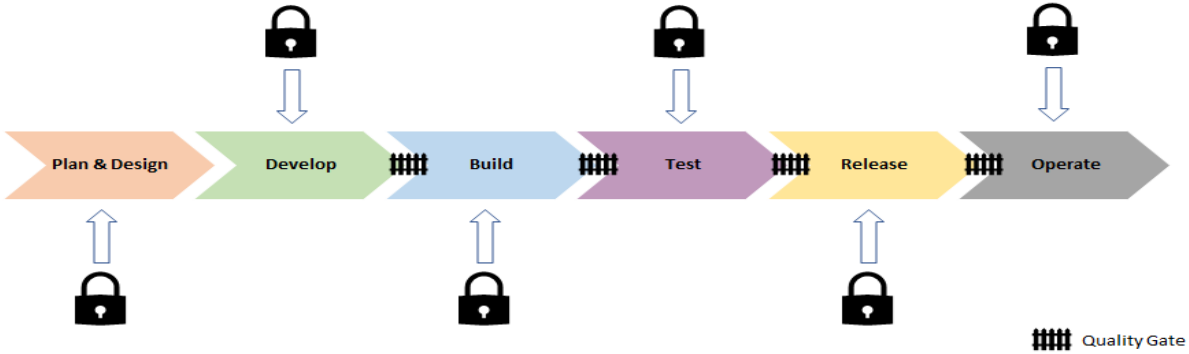
DevOps & Automation

Potential challenges:

- At this point, there may not be enough time in schedule to absorb change
- Activities required to remediate may be complex
- May require revisit of one or more previous phases to properly address

DevOps & Automation

Better approach:



DevOps & Automation

- Plan & Design – Threat modeling, data protection, and risk assessment
- Develop – SAST (Static Application Security Testing) tools
- Build – SAST and SCA (Software Composition Analysis) tooling
- Test – DAST (Dynamic Application Security Testing) tools, passive/active scans and “fuzzing”
- Release – Additional security-specific scanning, port scans, and log validation
- Operate – Monitoring & alerting, RCA (Root Cause Analysis)

DevOps & Automation

- Quality gates guard against moving security defects forward
- In true DevOps fashion:
  - Information gathered from early phases feeds into later phases
  - Lessons learned feed continuous improvement of overall process

Case Study – Discussion

Scenario: A global supplier is looking to modernize its order processing system. Among other things, this system includes Accounts Payable, Accounts Receivable, Inventory, Shipping, Invoicing, and Raw Materials modules. Most of their current system is built on a mainframe stack which is several versions behind in terms of system upgrades and patches, and, while the modules used by the company are provided with out-of-the-box implementations by the mainframe vendor, the company has chosen to heavily customize those modules over multiple years. Internal users interface with the modules of the system using "green screen" terminals. There is a thin web layer for presentation to and utilization by external users which submits Message Queueing (MQ) calls to the mainframe for the back-end business logic.

Let's discuss:

- Of the set of operational concerns reviewed on the previous slides, which do you think apply to this use case?
- Is there an operational concern (or smaller subset) which should be given higher priority for this system?

Lab 01 – Creating an EC2 Instance in AWS

Thank you!

If you have additional questions, please reach out to me at: (email address)

DevelopIntelligence
A PLURALSIGHT COMPANY