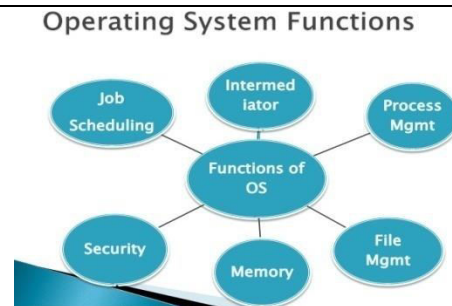
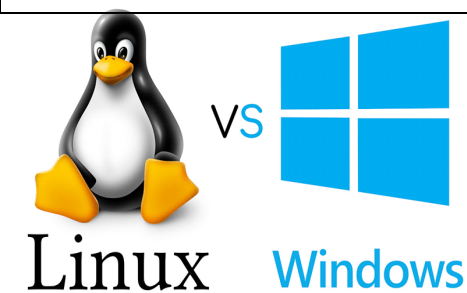


Week 10 Day 2: Cloud Implementation, AWS (EC2, S3, EBS, Beanstalk), Google Cloud Services, Google App Engine (GAE), Microsoft Azure, BLOBs, OpenStack, Cloud Controller



Major Cloud Providers

- **Amazon Web Services**
- **Microsoft**
- **Google**
- **Rackspace**
- **Virtustream**
- **CenturyLink**
- **VMware**
- **NTT Communication**
- **IBM(SoftLayer)**
- **Fujitsu**

What is Amazon Web Services ?

- Amazon Web Services (AWS) is a collection of remote computing services (web services) that together make up a cloud computing platform, offered over the Internet by Amazon.com.
- Website: <http://aws.amazon.com>

What is Amazon Web Services ?

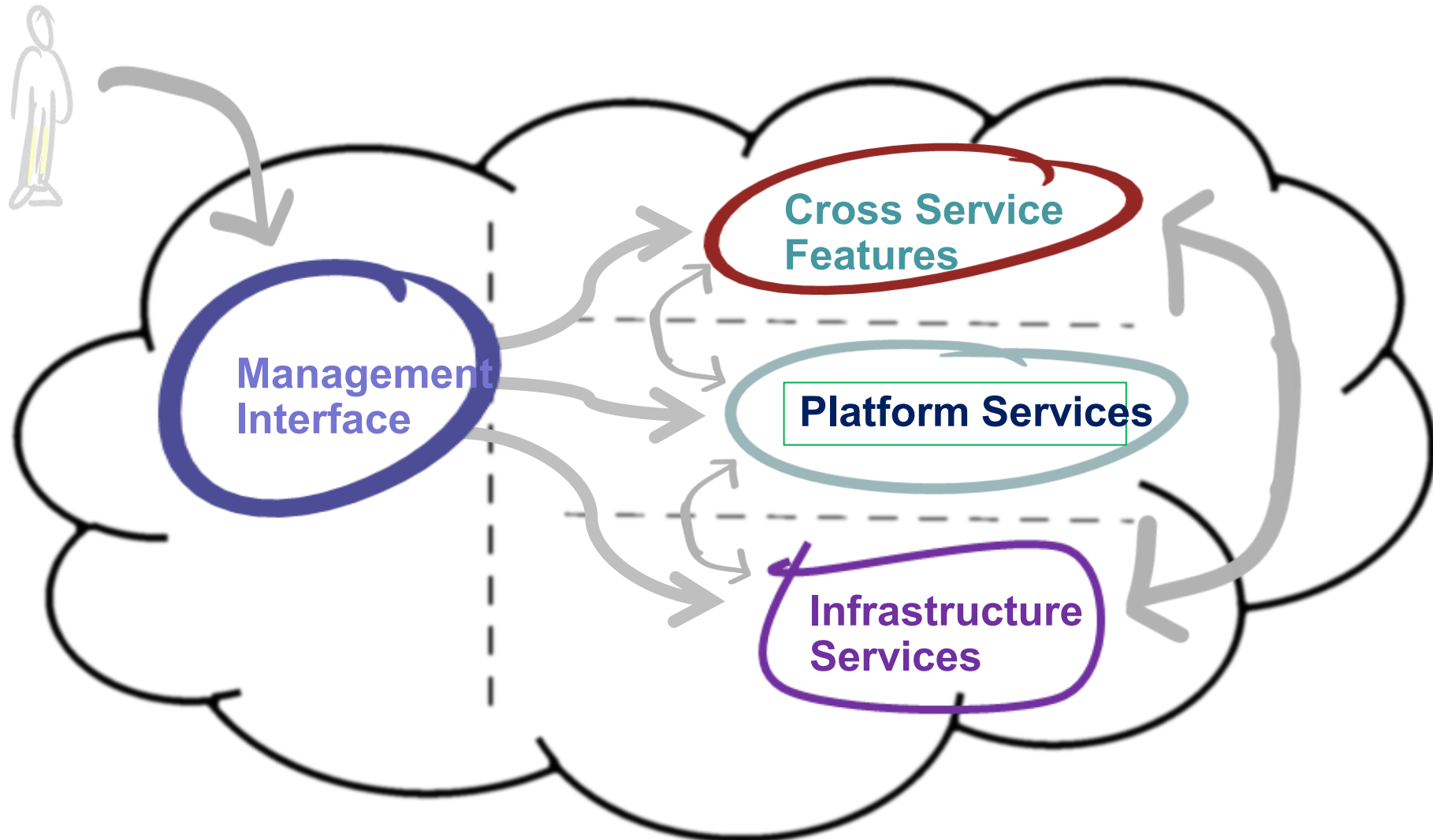
- AWS is located in 9 geographical 'Regions'. Each Region is wholly contained within a single country and all of its data and services stay within the designated Region.
- Each Region has multiple 'Availability Zones', which are distinct data centers providing AWS services. Availability Zones are isolated from each other to prevent outages from spreading between Zones. However, Several services operate across Availability Zones (e.g. S3, DynamoDB).



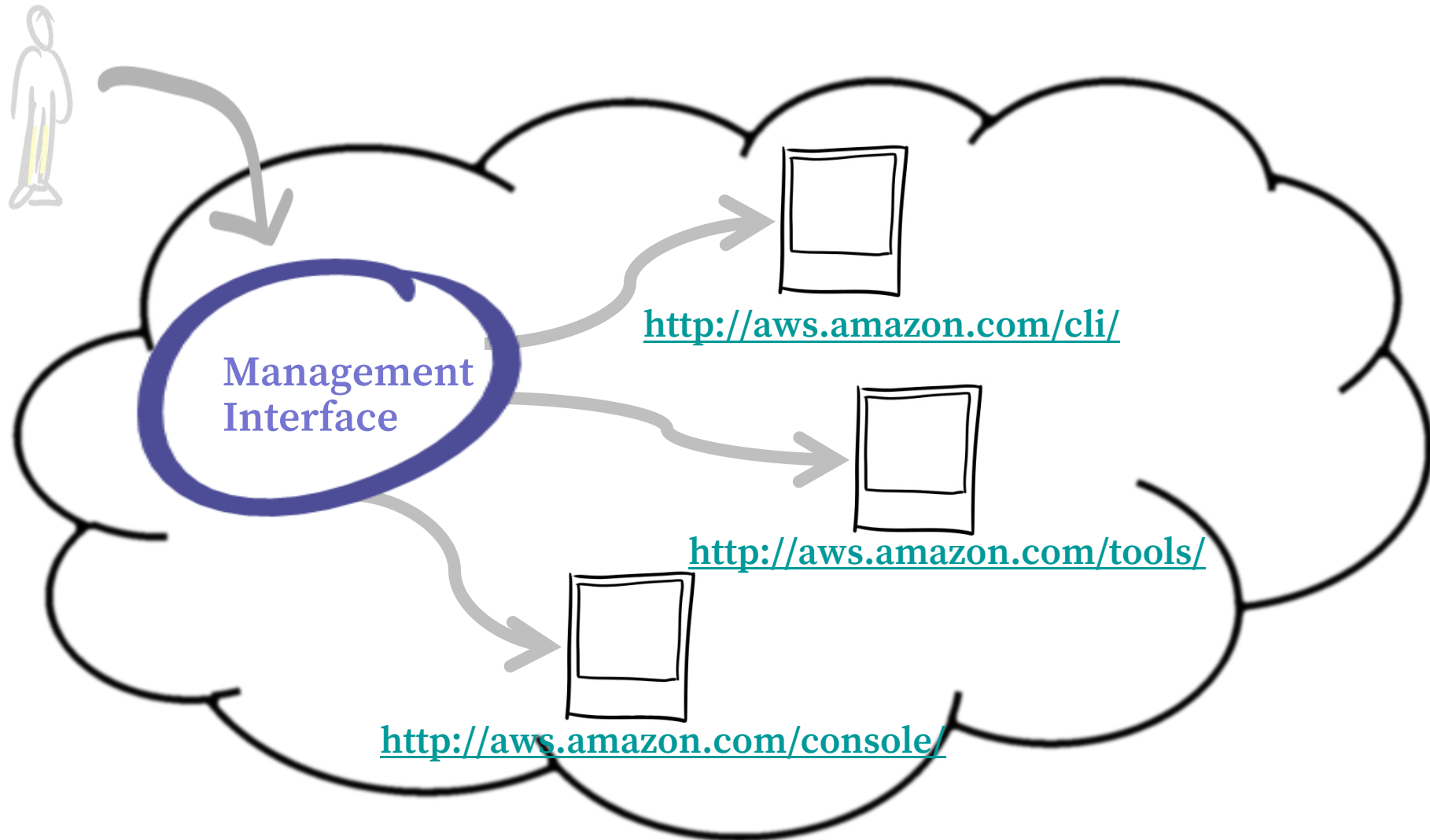
What is AWS Offering?

- **Low Ongoing Cost:**, pay-as-you-go pricing with no up-front expenses or long-term commitments.
- **Instant Elasticity & Flexible Capacity:** (scaling up and down) Eliminate guessing on your infrastructure capacity needs.
- **Speed & Agility:** Develop and deploy applications faster Instead of waiting weeks or months for hardware to arrive and get installed.
- **Apps not Ops:** Focus on projects. Lets you shift resources away from data center investments and operations and move them to innovative new projects.
- **Global Reach:** Take your apps global in minutes.
- **Open and Flexible:** You choose the development platform or programming model that makes the most sense for your business.
- **Secure:** Allows your application to take advantage of the multiple layers of operational and physical security in the AWS data centers to ensure the integrity and safety of your data.

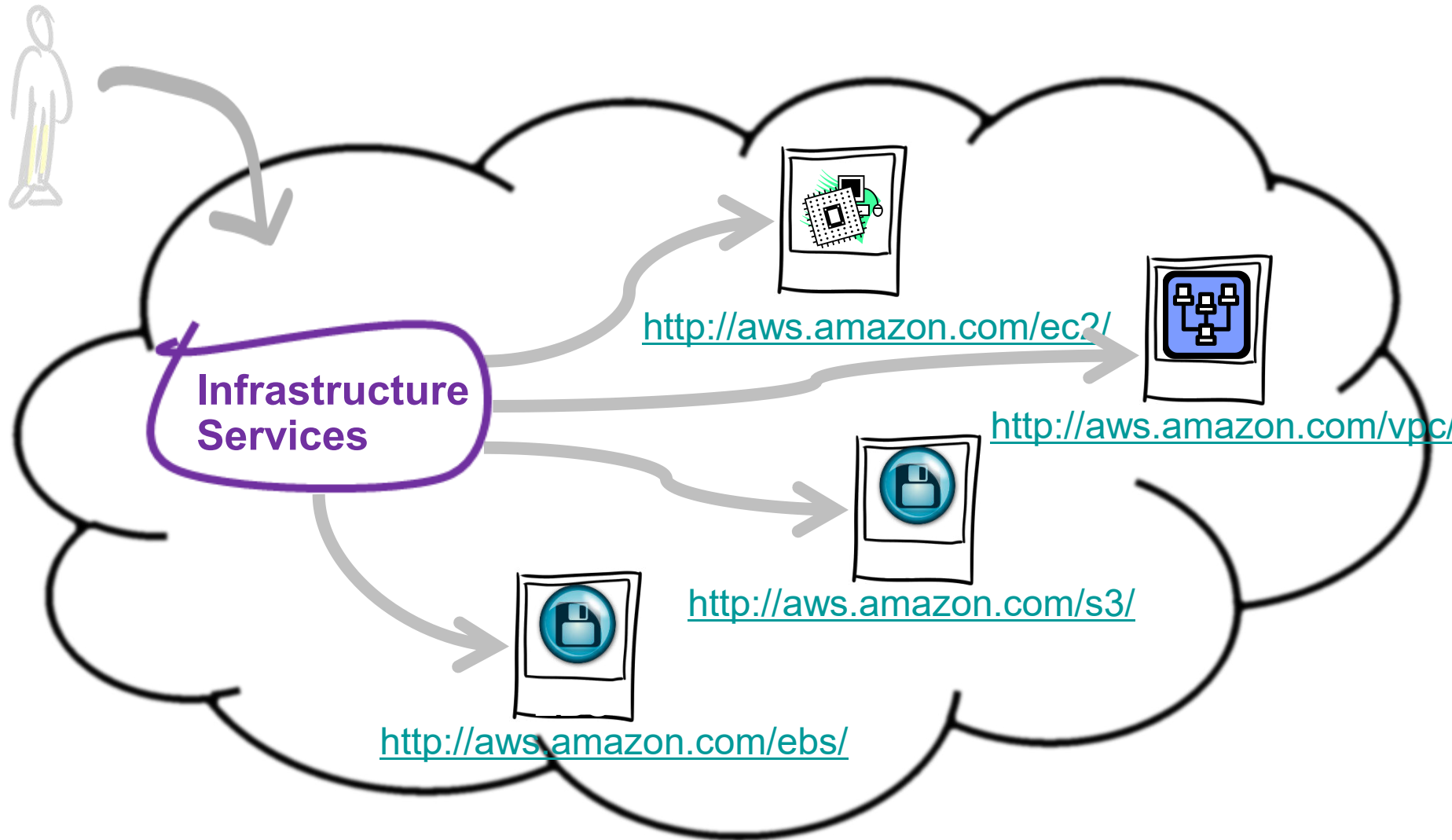
The Amazon Web Services Universe



Management Interface

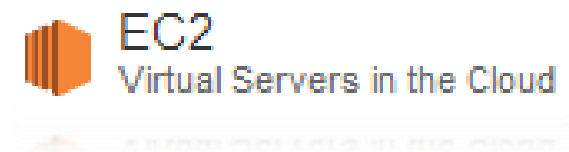


Infrastructure Services



Amazon Elastic Compute Cloud (EC2)

- A web service that provides **resizable compute capacity** in the cloud.
- EC2 allows **creating Virtual Machines (VM) on-demand**. Pre-configured **templated Amazon Machine Image (AMI)** can be used get running immediately. Creating and sharing your own AMI is also possible via the **AWS Marketplace**.
- **Auto Scaling** allows **automatically scale of the capacity up** seamlessly during **demand spikes** to maintain performance, and **scales down** during **demand lulls** to minimize costs.
- **Elastic Load Balancing** automatically distributes incoming application traffic across multiple Amazon EC2 instances.
- Provide tools to build **failure resilient applications** by launching application instances in **separate Availability Zones**.
- Pay only for resources actually consume, **instance-hours**.
- **VM Import/Export** enables you to easily import virtual machine images from your existing environment to Amazon EC2 instances and export them back at any time.



- **Micro instances (t1.micro):**

- Micro Instance 613 MiB of memory, up to 2 ECUs (for short periodic bursts), EBS storage only, 32-bit or 64-bit platform.

- **Standard Instances** provide customers with a balanced set of resources and a low cost platform.

- **M1 Small Instance (Default)** 1.7 GiB of memory, 1 EC2 Compute Unit (1 virtual core with 1 EC2 Compute Unit), 160 GB of local instance storage, 32-bit or 64-bit platform
- **M1 Medium Instance** 3.75 GiB of memory, 2 EC2 Compute Units (1 virtual core with 2 EC2 Compute Units each), 410 GB of local instance storage, 32-bit or 64-bit platform
- **M1 Large Instance** 7.5 GiB of memory, 4 EC2 Compute Units (2 virtual cores with 2 EC2 Compute Units each), 850 GB of local instance storage, 64-bit platform
- **M1 Extra Large Instance** 15 GiB of memory, 8 EC2 Compute Units (4 virtual cores with 2 EC2 Compute Units each), 1690 GB of local instance storage, 64-bit platform
- **M3 Extra Large Instance** 15 GiB of memory, 13 EC2 Compute Units (4 virtual cores with 3.25 EC2 Compute Units each), EBS storage only, 64-bit platform
- **M3 Double Extra Large Instance** 30 GiB of memory, 26 EC2 Compute Units (8 virtual cores with 3.25 EC2 Compute Units each), EBS storage only, 64-bit platform

One EC2 Compute Unit (ECU) provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or Xeon processor.

EC2 High Performance Instances

- **High-Memory Instances:**

- **High-Memory Extra Large Instance** 17.1 GiB memory, 6.5 ECU (2 virtual cores with 3.25 EC2 Compute Units each), 420 GB of local instance storage, 64-bit platform
- **High-Memory Double Extra Large Instance** 34.2 GiB of memory, 13 EC2 Compute Units (4 virtual cores with 3.25 EC2 Compute Units each), 850 GB of local instance storage, 64-bit platform
- **High-Memory Quadruple Extra Large Instance** 68.4 GiB of memory, 26 EC2 Compute Units (8 virtual cores with 3.25 EC2 Compute Units each), 1690 GB of local instance storage, 64-bit platform

- **High-CPU Instances**

- **High-CPU Medium Instance** 1.7 GiB of memory, 5 EC2 Compute Units (2 virtual cores with 2.5 EC2 Compute Units each), 350 GB of local instance storage, 32-bit or 64-bit platform
- **High-CPU Extra Large Instance** 7 GiB of memory, 20 EC2 Compute Units (8 virtual cores with 2.5 EC2 Compute Units each), 1690 GB of local instance storage, 64-bit platform

EC2 High Performance Instances

- **High Storage Instances**

- **High Storage Eight Extra Large** 117 GiB memory, 35 EC2 Compute Units, **24 * 2 TB of hard disk drive local instance storage**, 64-bit platform, 10 Gigabit Ethernet

- **High I/O Instances**

- **High I/O Quadruple Extra Large** 60.5 GiB memory, 35 EC2 Compute Units, **2 * 1024 GB of SSD-based local instance storage**, 64-bit platform, 10 Gigabit Ethernet

EC2 Cluster Instances

- **Cluster Compute Instances** provide proportionally high CPU resources with increased network performance and are well suited for High Performance Compute (HPC) applications and other demanding network-bound applications.
 - **Cluster Compute Eight Extra Large** 60.5 GiB memory, 88 EC2 Compute Units, 3370 GB of local instance storage, 64-bit platform, 10 Gigabit Ethernet
- **High Memory Cluster Instances** provide proportionally high CPU and memory resources with increased network performance, and are well suited for memory-intensive applications including in-memory analytics, graph analysis, and scientific computing.
 - **High Memory Cluster Eight Extra Large** 244 GiB memory, 88 EC2 Compute Units, 240 GB of local instance storage, 64-bit platform, 10 Gigabit Ethernet

EC2 Cluster Instances

- **Cluster GPU Instances** provide general-purpose graphics processing units (GPUs) with proportionally high CPU and increased network performance for applications benefitting from highly parallelized processing, including HPC, rendering and media processing applications.
 - **Cluster GPU Quadruple Extra Large** 22 GiB memory, 33.5 EC2 Compute Units, 2 x NVIDIA Tesla “Fermi” M2050 GPUs, 1690 GB of local instance storage, 64-bit platform, 10 Gigabit Ethernet.

EC2 Payment methods

- **On-Demand Instances** let you pay for compute capacity by the hour with no long-term commitments.
- **Reserved Instances** give you the option to make a low, one-time payment for each instance you want to reserve and in turn receive a significant discount on the hourly charge for that instance.
- **Spot Instances** allow customers to bid on unused Amazon EC2 capacity and run those instances for as long as their bid exceeds the current Spot Price.

Amazon Elastic Block Store (EBS)

- Provides **block level storage volumes (1 GB to 1 TB)** for use with **Amazon EC2 instances**.
 - **Multiple volumes** can be mounted to the **same instance**.
 - EBS volumes are **network-attached**, and **persist independently** from the life of an instance.
 - Storage volumes behave like **raw, unformatted block devices**, allowing users to **create a file system** on top of Amazon EBS volumes, or use them in any other way you would use a block device (**like a hard drive**).
- EBS volumes are **placed in a specific Availability Zone**, and can then be **attached to instances also in that same Availability Zone**.
- Each storage volume is **automatically replicated within the same Availability Zone**.
- EBS provides the ability to **create point-in-time snapshots of volumes**, which are **persisted to Amazon S3**.
 - These snapshots can be **used as the starting point for new Amazon EBS volumes**, and protect data for long-term durability.
 - The **same snapshot can be used to instantiate as many volumes as you wish**.
 - These snapshots can be copied across **AWS regions**.

Infrastructure
Services

EBS Volumes

- **Standard volumes** offer storage for applications with **moderate or bursty I/O** requirements.

- Standard volumes deliver approximately **100 IOPS** on average.
- well suited for use as **boot volumes**, where the burst capability provides fast instance start-up times.

- **Provisioned IOPS volumes** are designed to deliver **predictable, high performance for I/O intensive workloads such as databases**.

- You specify an **IOPS rate** when creating a volume, and **EBS** provisions that rate for the lifetime of the volume.
- Amazon EBS currently supports **up to 4000 IOPS per Provisioned IOPS volume**.
- You can **stripe multiple volumes together** to deliver thousands of IOPS per EC2 instance.

- To enable your EC2 instances to fully utilize the IOPS provisioned on an EBS volume,:

- Launch selected Amazon EC2 instance types as “**EBS-Optimized**” instances.
- **EBS-optimized instances deliver dedicated throughput between Amazon EC2 and Amazon EBS**, with options between **500 Mbps and 1000 Mbps** depending on the instance type used.

- **EBS charges based on per GB-month AND per 1 million I/O requests**

Amazon Simple Storage Service (S3)

- Amazon S3 provides a simple web services interface that can be used to store and retrieve **any amount of data, at any time, from anywhere** on the web.
- Write, read, and delete objects containing from **1 byte to 5 terabytes** of data each. The **number of objects** you can store is **unlimited**.
- Each object is stored in a **bucket** and retrieved via a unique, developer-assigned **key**.
 - A bucket can be stored in **one of several Regions**.
 - You can **choose a Region** to optimize for latency, minimize costs, or address regulatory requirements.
 - Objects stored in a Region **never leave the Region** unless you transfer them out.
- **Authentication mechanisms** are provided to ensure that data is kept secure from unauthorized access.
 - Objects can be made **private or public**, and rights can be granted to specific users.
- S3 charges based on **per GB-month AND per I/O requests AND per data modification requests**.



S3

Scalable Storage in the Cloud

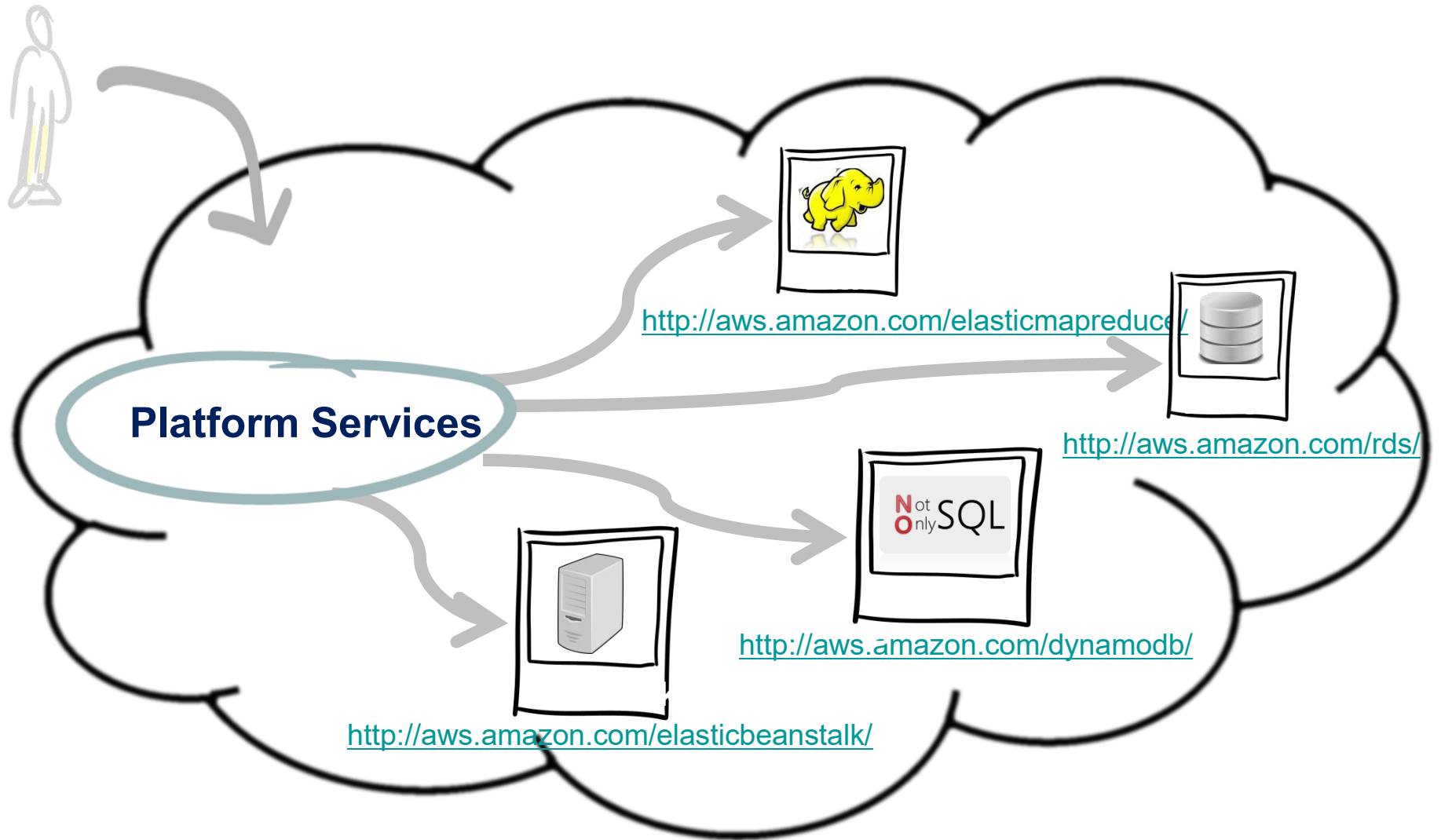
Amazon Virtual Private Cloud (VPC)

- **Amazon VPC** lets you provision a logically isolated section of the Amazon Web Services (AWS) Cloud.
- You have **complete control** over your virtual networking environment, including:
 - selection of your own **IP address range**,
 - **creation of subnets**, and
 - **configuration of route tables and network gateways**.
- VPC allows bridging with an onsite IT infrastructure with an **encrypted VPN connection** with an **extra charge per VPN Connection-hour**.
- There is **no additional charge** for using Amazon Virtual Private Cloud, aside from the normal Amazon EC2 usage charges.

Infrastructure
Services

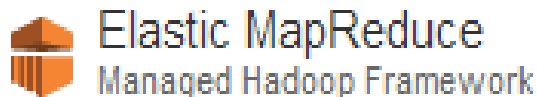


Platform Services



Amazon Elastic MapReduce (EMR)

- Amazon EMR is a web service that makes it easy to quickly and cost-effectively process vast amounts of data using Hadoop.
- Amazon EMR distribute the data and processing across a resizable cluster of Amazon EC2 instances.
- With Amazon EMR you can launch a persistent cluster that stays up indefinitely or a temporary cluster that terminates after the analysis is complete.
- Amazon EMR supports a variety of Amazon EC2 instance types and Amazon EC2 pricing options (On-Demand, Reserved, and Spot).
- When launching an Amazon EMR cluster (also called a "job flow"), you choose how many and what type of Amazon EC2 Instances to provision.
- The Amazon EMR price is in addition to the Amazon EC2 price.
- Amazon EMR is used in a variety of applications, including log analysis, web indexing, data warehousing, machine learning, financial analysis, scientific simulation, and bioinformatics.



Platform Services

Amazon Relational Database Service (RDS)

- **Amazon RDS** is a web service that makes it easy to set up, operate, and scale a relational database in the cloud.
- **Amazon RDS** gives access to the capabilities of a familiar **MySQL, Oracle or Microsoft SQL Server** database engine.
 - Code, applications, and tools already used with existing databases can be used with RDS.
- **Amazon RDS** automatically patches the database software and backs up the database, storing the backups for a user-defined retention period and enabling point-in-time recovery.
- **Amazon RDS** provides scaling the compute resources or storage capacity associated with the Database Instance.
- **Pay** only for the resources actually consumed, based on the DB Instance hours consumed, database storage, backup storage, and data transfer.
 - **On-Demand DB Instances** let you pay for compute capacity by the hour with no long-term commitments.
 - **Reserved DB Instances** give the option to make a low, one-time payment for each DB Instance and in turn receive a significant discount on the hourly usage charge for that DB Instance.

Platform Services



RDS

Managed Relational Database Service

- In relational databases (SQL Databases), **ACID (Atomicity, Consistency, Isolation, Durability)** is a set of properties that guarantee that database transactions are processed reliably.
 - **Atomicity** requires that each transaction is "all or nothing": if one part of the transaction fails, the entire transaction fails, and the database state is left unchanged.
 - The **consistency** property ensures that any transaction will bring the database **from one valid state to another**.
 - The **isolation** property ensures that the **concurrent execution of transactions** results in a system state that would be obtained if transactions were executed **serially**,.
 - **Durability** means that **once a transaction has been committed, it will remain so**, even in the event of power loss, crashes, or errors.

Amazon DynamoDB

- **DynamoDB** is a fast, fully managed **NoSQL database service** that makes it simple and cost-effective to store and retrieve any amount of data, and serve any level of request traffic.
- All data items are stored on **Solid State Drives (SSDs)**, and are **replicated** across **3 Availability Zones** for high availability and durability.
- **DynamoDB tables do not have fixed schemas**, and each item may have a **different number of attributes**.
- **DynamoDB has no upfront costs** and implements a **pay as you go** plan as a. a flat hourly rate based on the **capacity reserved**.

Platform Services



DynamoDB

Predictable and Scalable NoSQL Data Store

NoSQL Databases

- A **NoSQL database** provides a mechanism for storage and retrieval of data that **employs less constrained consistency models** than traditional relational databases.
- NoSQL databases only support **Eventual Consistency** which is a consistency model used in distributed computing that **informally guarantees** that, if no new updates are made to a given data item, **eventually all accesses to that item will return the last updated value**.
- NoSQL databases are often **highly optimized key-value stores** intended for **simple retrieval and appending operations**, with the goal being significant **performance benefits** in terms of **latency and throughput**.
- **Key-value stores** allow the application to store its data in a **schema-less way**.
 - The data could be stored in a **datatype** of a programming language or an object. Because of this, there is **no need for a fixed data model**.

Amazon Elastic Beanstalk

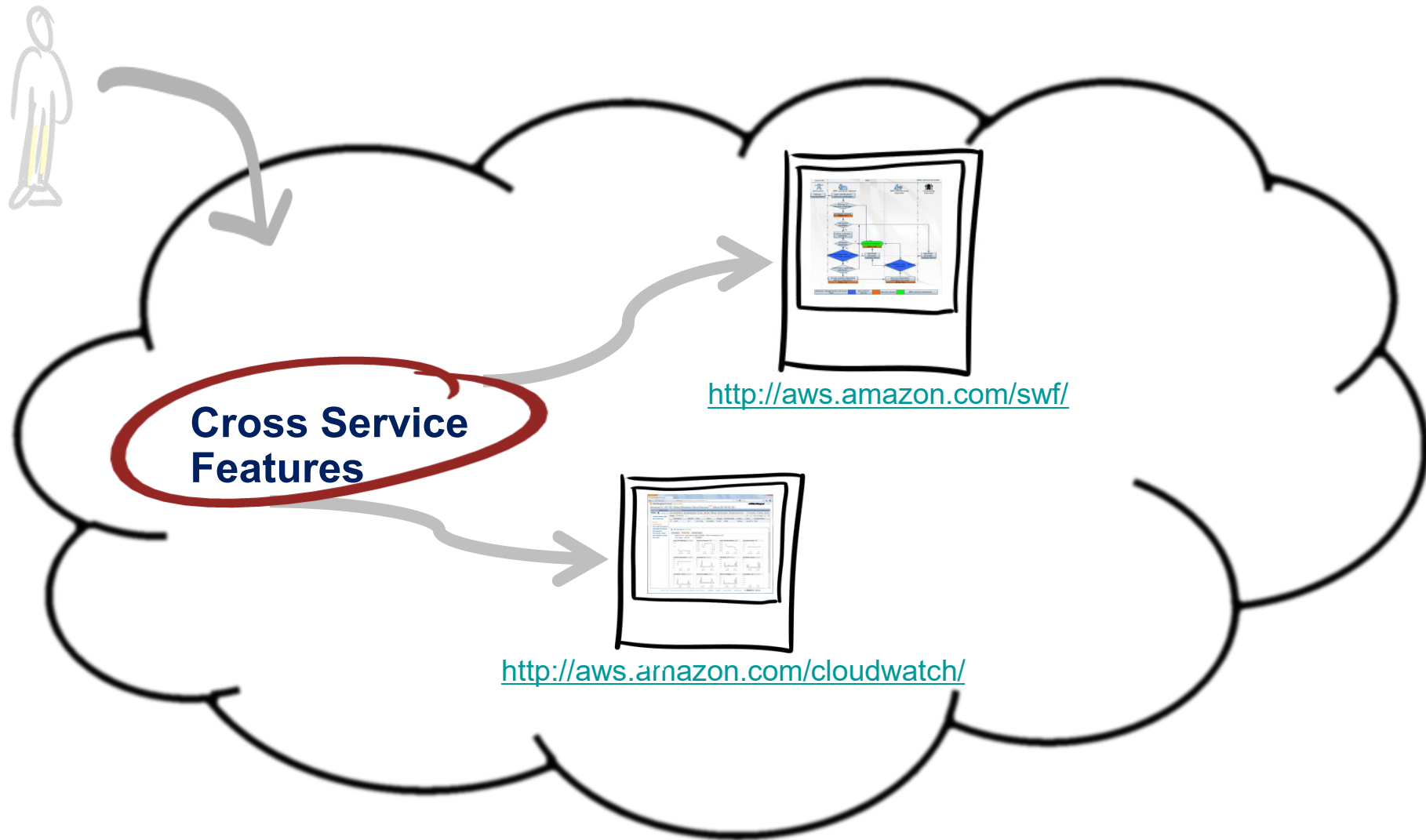
- **AWS Elastic Beanstalk** provides a solution to **quickly deploy and manage** applications in the AWS cloud.
- You **simply upload your application**, and Elastic Beanstalk **automatically** handles the deployment details of capacity **provisioning**, **load balancing**, **auto-scaling**, and application health **monitoring**.
- Elastic Beanstalk leverages AWS services such as **Amazon EC2**, **Amazon S3**,
- To ensure easy portability of your application, Elastic Beanstalk is built using familiar software stacks such as:
 - Apache HTTP Server for Node.js, PHP and Python
 - Passenger for Ruby,
 - IIS 7.5 for .NET
 - Apache Tomcat for Java.
- There is **no additional charge** for Elastic Beanstalk - you **pay only for the AWS resources** needed to store and run your applications.

Platform Services



Elastic Beanstalk
AWS Application Container

Cross Service Features



Amazon CloudWatch

- Amazon **CloudWatch** provides **monitoring** for **AWS** cloud resources and the applications customers run on **AWS**.
- Amazon CloudWatch lets you **programmatically** retrieve your **monitoring data**, view **graphs**, and **set alarms** to help you troubleshoot, spot trends, and **take automated action** based on the state of your cloud environment.
- Amazon CloudWatch enables you to monitor your **AWS** resources **up-to-the-minute in real-time**, including:
 - Amazon EC2 instances,
 - Amazon EBS volumes,
 - Elastic Load Balancers,
 - Amazon RDS DB instances.
- Metrics such as **CPU utilization**, **latency**, and **request counts** are provided automatically for these **AWS** resources.
- Customers can also supply their **own custom application and system metrics**, such as **memory usage**, **transaction volumes**, or **error rates**.

Cross Service
Features



CloudWatch
Resource and Application Monitoring

Amazon Simple Workflow Service (SWF)

- Amazon SWF is a **task coordination** and **state management service** for cloud applications.
- Using Amazon SWF, you **structure** the various **processing steps** in an application that **runs across one or more machines** as a set of “**tasks**.”
- Amazon SWF **manages dependencies** between the tasks, **schedules** the tasks for execution, and runs any logic that needs to be **executed in parallel**.
- The service also **tracks** the tasks’ **progress**.
- As the **business requirements change**, Amazon SWF makes it **easy to change application logic** without having to worry about the **underlying state machinery** and **flow control**.

Cross Service
Features



SWF

Workflow Service for Coordinating Application Components

Watch out for unexpected Costs

- When you finish your work remember to make sure of the following to **avoid unwanted costs**:
 - **Delete** your **S3** objects.
 - **Stop** or **Shut Down** your **EC2** and **RDS** instances.
- The customer is responsible for the resources he's using. **AWS declines any responsibility** if the customer forgets to shut down resources.

- IaaS & PaaS services
- All services built on top of Google hardware
- located across the globe
- Global redundancy and service locality
- Utilizes Google's Global network
 - Dedicated fiber optic networks
 - Even between continents
 - Multilevel location based data caching
 - Google Managed Services
- – Database administration, server configuration , load balancing managed for the user
- <https://cloud.google.com/>

- Compute Engine
 - IaaS providing virtual machines similar to Amazon EC2
- App Engine
 - PaaS for directly hosting applications
- Container Engine
 - Powerful cluster manager and orchestration system for running your Docker containers
- Docker uses the resource isolation features of the Linux kernel such as cgroups and kernel namespaces
- Independent "containers" run within a single Linux instance
- Avoids the overhead of starting and maintaining virtual machines

Google Cloud Services - Storage

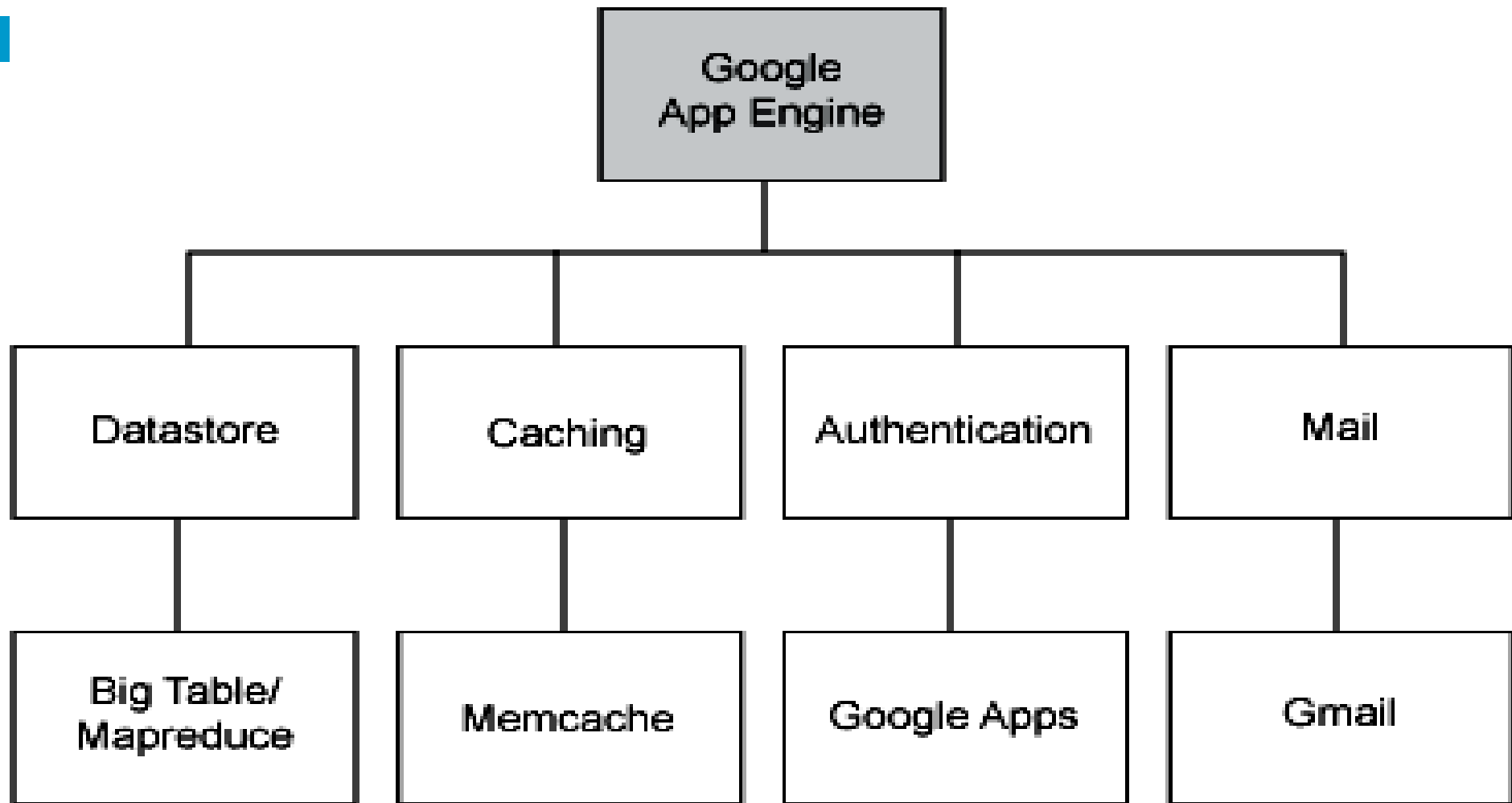
- Cloud Storage
 - Object storage for applications
- Cloud SQL
 - Fully managed Relational MySQL
 - beta support for PostgreSQL
- Cloud Datastore
 - Automatically scaled NoSQL storage
- Cloud BigTable
 - NoSQL database for warehousing
 - A high performance NoSQL database service for large analytical and operational workloads

Google Cloud Services – (continued)

- Big Data
 - Big Query
- SQL like queries against multi-terabyte/petabyte datasets
 - Cloud Dataflow
- Managing data processing workflows
- Supports both stream and batch execution of pipelines
 - Cloud DataProc
- Managed data processing with Hadoop and Spark
- Cloud Endpoints
 - Create RESTful services to make your code accessible from iOS, Android and Javascript clients

- PaaS for developing and hosting web applications in Google-managed data centers
- Easy to build, maintain, and scale applications
- No servers to maintain or configure by yourself
- Upload & Go
- Was created before Google Cloud became available for public use
- Supported languages
 - Python, Java, PHP, Go

What does GAE Provide



App Engine Characteristics

- **Persistent storage** with queries, sorting, and transactions
- App Engine distributes user requests across multiple servers and **scales servers** to meet **dynamic traffic** demands
- **Asynchronous task queues** for performing work outside the scope of a request

App Engine Characteristics

- **Scheduled tasks** for triggering events at specified times or regular intervals
- **Integration** with all other Google Cloud services and APIs
- Your application runs within its own **secure, sandboxed and reliable** environment
 - Independent of the hardware, operating system, or physical location of the server

Services of Relevance for App Engine

- **Google Cloud SQL** - A fully-managed web service that allows you to create, configure, and use relational databases in Google's cloud
- **Datastore** - A schemaless object datastore providing robust, scalable storage for your web application, a rich data modeling API, and a SQLlike query language called GQL
- **Blobstore** - Allows your application to serve large data objects, such as video or image files, that are too large for storage in the Datastore service

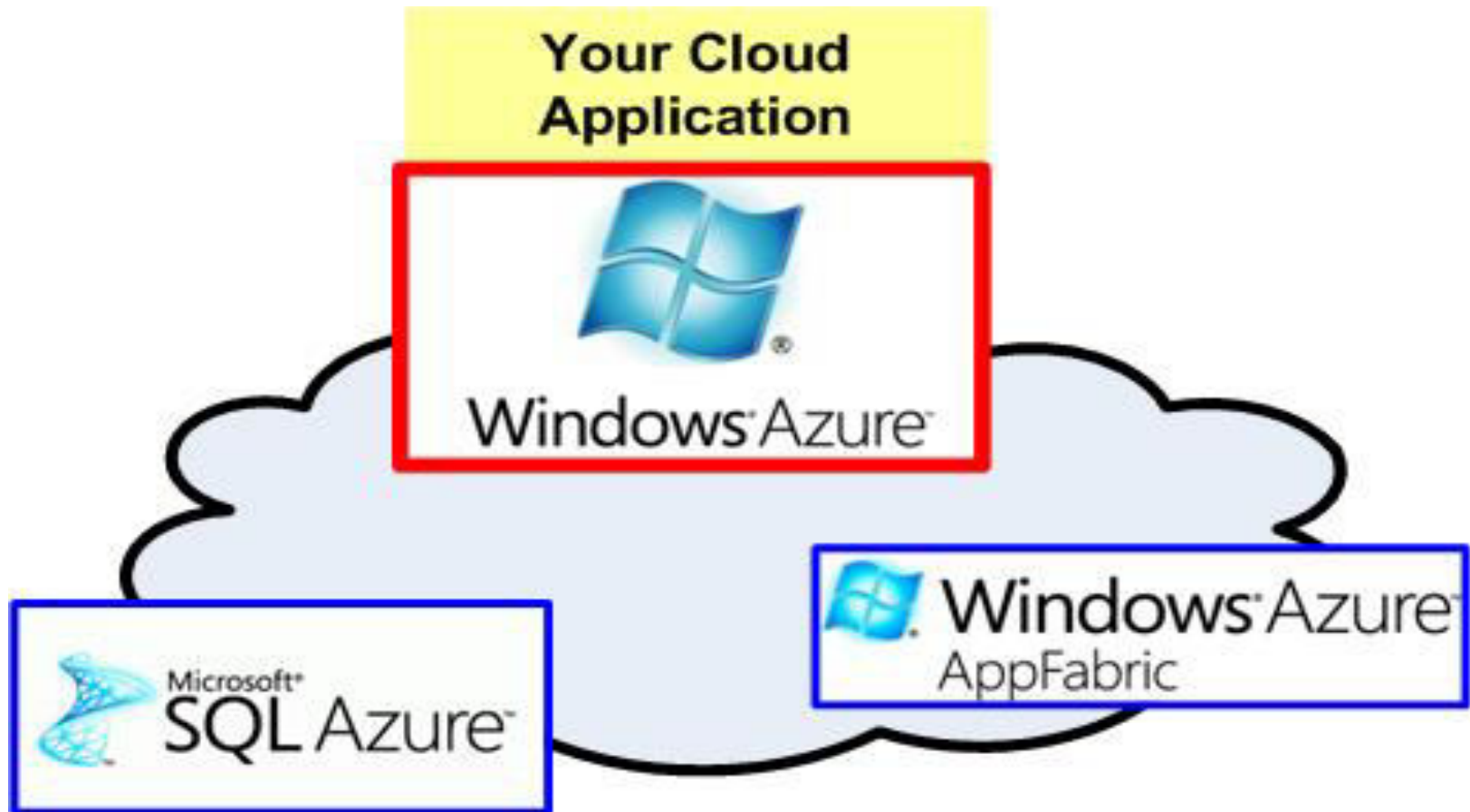
Other data services

- **Search** - Allows your application to perform Google-like searches over structured data such as: plain text, HTML, atom, numbers, dates, and geographic locations.
- **Memcache** - A distributed, in-memory data cache to improve application performance
- **Logs** - Provides programmatic access to application and request logs from within your application

- **Channel** - Creates a persistent connection between your application in Google servers and JavaScript clients so you can send messages to clients in real time without "polling"
- **Mail** - Sends email messages on behalf of administrators and users with Google Accounts, and receives mail at various addresses
- **XMPP** - Enables an application to send and receive chat messages to and from any XMPP-compatible chat messaging service
- **Traffic Splitting** - Allows you to roll out features for your app slowly over a period of time. Traffic Splitting works by splitting incoming requests to different versions of your app

- **Task Queue** - Allows applications to perform work outside of a user request, and organize that work into small, discrete units, called "tasks," to be executed later
- **Scheduled Tasks** - Allows applications to configure regularly scheduled tasks that operate at defined times or regular intervals
- **Remote** - Lets external applications transparently access App Engine services. For example, you can use Remote API to access a production datastore from an app running on your local machine.

- **App Identity** - Gives code access to the application identity; provides framework to assert this identity over OAuth.
- **Users** - Allows applications to sign in users with Google Accounts or OpenID, and address these users with unique identifiers.
- **Capabilities** - Provides detection of outages and scheduled maintenance for specific APIs and services, so that your application may bypass them or inform your users.
- **Multitenancy** - Makes it easy to compartmentalize your data to serve many client organizations from a single instance of your application.
- **PageSpeed** - A family of tools that automatically optimizes the performance of your application.



- Hybrid PaaS & IaaS cloud platform
- Designed more for enterprise applications
- Programming languages
 - .NET, Java, PHP, Node.js, Python, or Ruby
- Datastores
 - Azure SQL database and NoSQL storage

<https://azure.microsoft.com>

What is an Azure application?

A service must include at least one role of either type

Web role is frontend, Worker Role is backend

Web role is worker role with IIS installed

WEB ROLE

- Hosted IIS
- HTTP/HTTPS
- ASP.NET
- Fast CGI + PHP

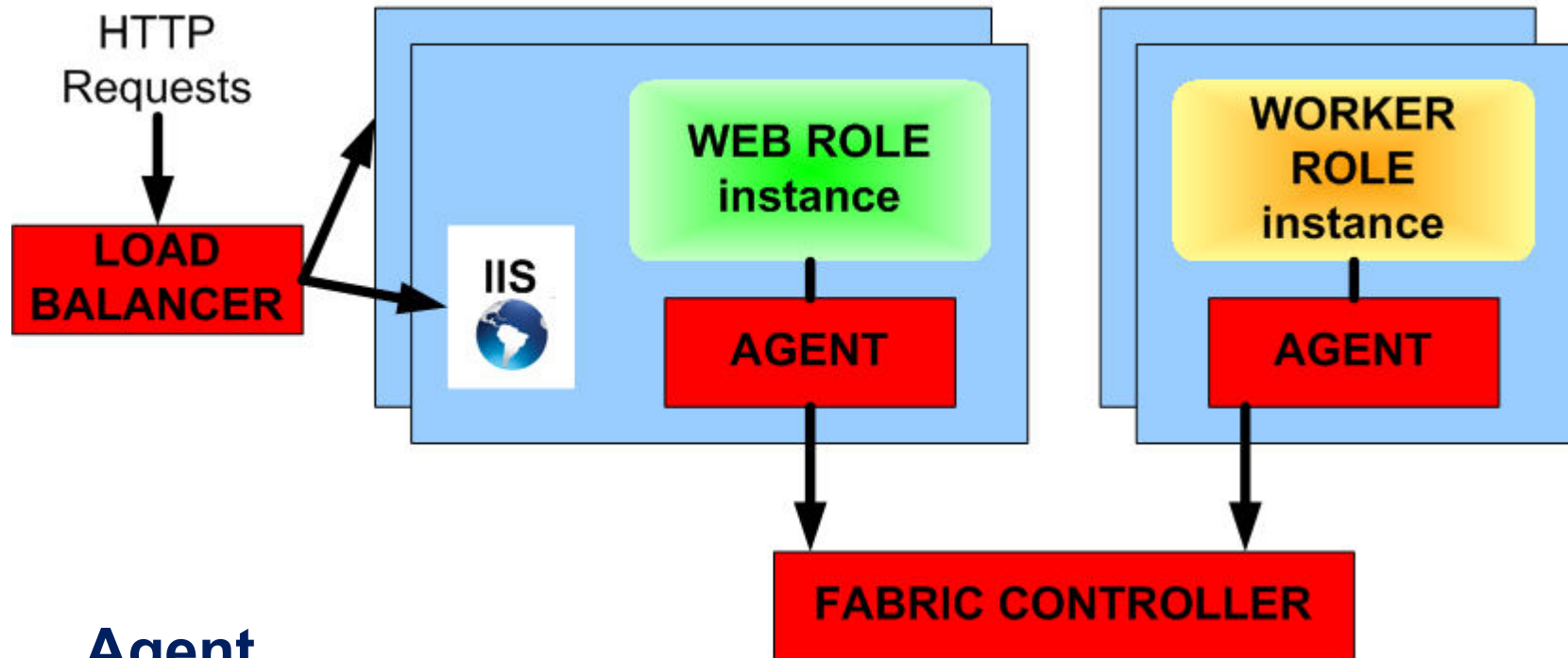
WORKER ROLE

- Managed Code Start
- Inbound on
 - Any TCP Port
 - HTTP/HTTPS

Application

- **Web Roles and Worker Roles**
- **Stateless design**
 - Easy-to-Scale
 - Fault Tolerance and Recovery
- **Under-the-cover Multiple instances**
 - Each runs in Microsoft Virtual Machine
 - Handled automatically by hypervisor

Agent and Fabric



Agent

- Exposes the API
- Monitors the failure conditions of the application

Fabric

- Allocate resources according to configuration file
- Detect and restart failed web roles and workers

Windows Azure Storage

Scalable storage in the cloud

- 100 TB per storage account
- Auto-scale to meet massive volume and throughput

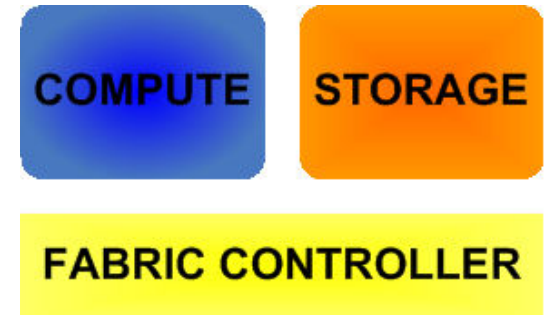
4 Types: Tables, Queues, BLOBs, Drives

Accessible via RESTful Web Service API

- Access from Windows Azure Compute
- Access from anywhere via internet
- Support .NET client library

Features

- Fault-Tolerance: All data replicated 3 times
- Guaranteed consistency
- Globally Visible: Accessible by non-Azure apps



- SQL Azure Database
 - Cloud version of Microsoft SQL Server
 - Indexes, views, stored procedures, triggers, etc..
 - SQL Server Reporting Services can hook into it
- Similar in usage to SQL Server
 - SQL semantics (CREATE, SELECT, UPDATE, etc)
- Difference in administration
 - Cannot control physical resources allocation
 - Automatic fail-over, replication

Key Benefits

- Self-Managing
 - No-hassle management
- High Availability
 - Replication
 - Automatic fail-over from hardware failures
- Scalability
 - Simply increase data storage in config
- Familiar and Standard Data Model
 - SQL, relational

Not to be confused with Window Azure's Fabric Controller

Connecting distributed applications

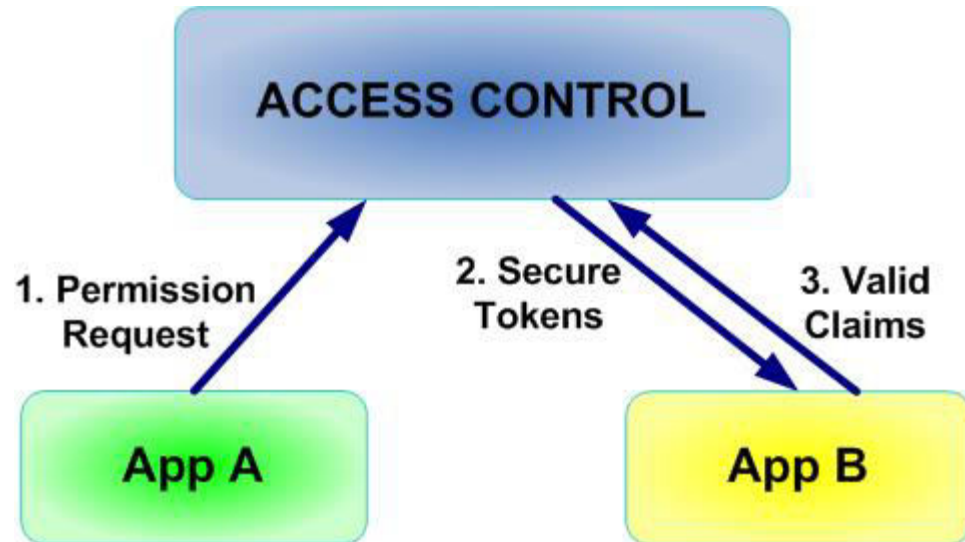
Service Bus

Access Control

Access Control

Claim-based Security Model

- Client must present token for action
 - Security token for “listen” claim
 - Security token for “send” claim
- Clients and Services supply credentials to Access Control to acquire security tokens
- Access Control has GUI for managing rules for issuing claims



- Storage characteristics
 - Durable – replicated three times
 - Scalable (capacity and throughput)
 - Highly available
- Simple and familiar programming interfaces
 - REST (HTTP and HTTPS)
 - .NET accessible

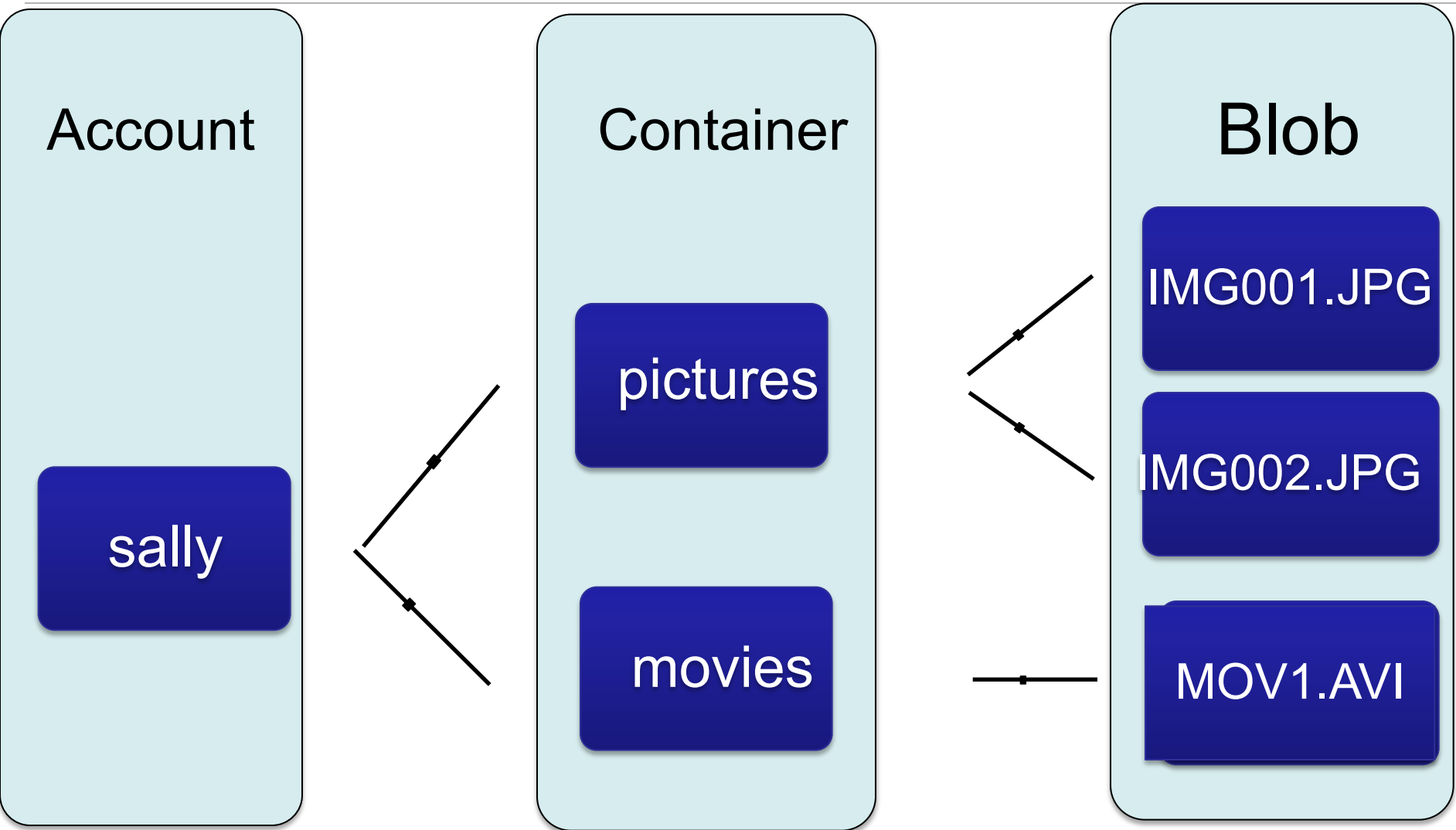
Windows Azure Storage Data Abstractions

- **Blobs** – File system in the cloud
- **Tables** – Massively scalable structured storage
- **Queues** – Reliable storage and delivery of messages
- **Drives** – Durable NTFS volumes for Windows Azure applications

BLOBs (Basic Large Objects)

- Intended for unstructured data
- Containers and BLOBs
 - Each Azure account has containers
 - Containers have one or more BLOBs.
 - BLOBs can be as large as terabyte

Blob Storage Concepts



- Mechanism for viewing persistent storage as if it were a local drive
- BLOBs can be exposed via Xdrives
- A mounted drive on a BLOB
- NTFS VHD mounted into Compute instance
- Cannot be remotely mapped

Tables

- Structured data but not quite relational
- Table has a collection of entities
- Entities are similar to 'rows'
- Entity has a primary key and properties (set of name/value pairs)
- E.g. an Entity of GuestBook which contains the guest user's information
 - ["GuestName"] = Joe Smith
 - ["Address"] = 30 Rockefeller Plaza, New York, NY
- In addition, Entity has PartitionKey and Timestamp
 - PartitionKey for load balancing

- **Transmission Problems with BLOBs?**
 - Divided into blocks for efficient transfer and retransmissions
- **Finding information?**
 - BLOBs can have associated metadata (e.g., photo)
- **Fast access?**
 - **Geolocation via content delivery network**
 - Storing frequently accessed data closer to apps that use it

- Free open-source cloud-computing software platform
- IaaS
- Began 2010 Rackspace and NASA
 - Early code from NASA's Nebula platform and Rackspace's Cloud Files platform
- Red Hat – commercial support with Grizzly
- Managed by OpenStack Foundation
 - Non-profit corporate entity
- AWS compatibility

What is with all the names?

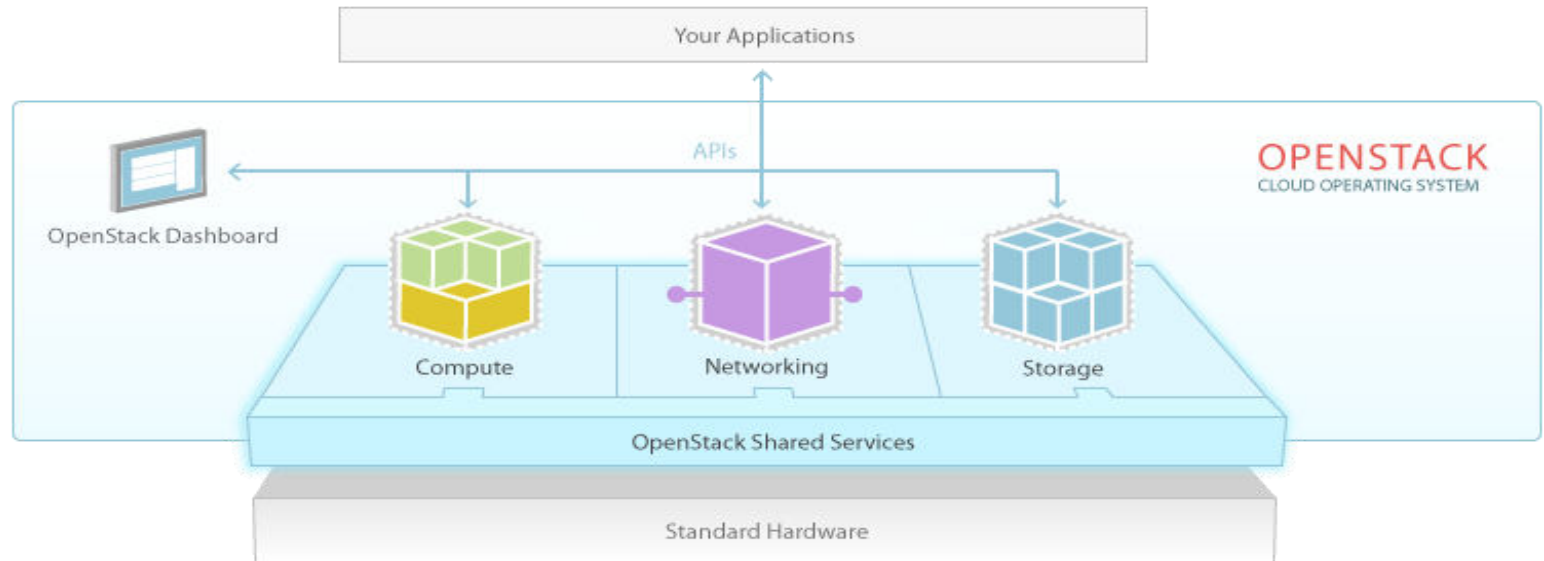
- All different flavors(?)
- Versions have different names – current release is Newton
Previous was kilo
- HP Helion has tweaked the storage of open stack
- Mirantis openstack allows companies to write their own plug-ins that can be installed and deployed automatically
- GroundWork supports all and added CloudHub

In general clouds have

- Controller node
- Compute node
- Storage node

- General components in the architecture

- Cloud Controller
- Compute Nodes – Nova
- Network Controller – Neutron
- Storage Controller – Swift



Core Services

- Nova – compute instance management for their lifecycle
- Keystone – identity provides authentication and authorization for other OpenStack services
- Glance – Image service to store/retrieve VMIs
- Neutron – network connectivity as a service for other OpenStack services
- Swift – object storage for unstructured data objects
- Cinder – block storage for running instances

Cloud controller

- Cloud controller is one or more nodes, that control the cloud operations.
- In contrast to compute nodes (e.g. worker nodes), cloud controller maintains a high level view of the resources and provides unified point for cloud management.
- All of the user's request (e.g. launch an instance), goes into cloud controller node and then it sends to the other nodes (e.g. compute nodes)
- In most non-production OpenStack environment, one controller node is enough.
- To achieve better high availability, two or more cloud controllers can be deployed.

OpenStack Cloud controller

- Generally, OpenStack Cloud Controller comprises of the following services:
- Database Service(MySQL/MariaDB): Stores cloud data
- Messaging Queue Service(RabbitMQ): Provides message passing mechanism for cloud nodes
- Authentication Service(Keystone): Provides authentication and authorization service
- API Endpoints(nova-api, glance-api, ...): Provides unified access point for cloud services(e.g. network, storage, compute)
- Scheduler(nova-scheduler, cinder-scheduler, ...): Provides workload scheduling service
- Dashboard(Horizon): Provides web-based management dashboard

- Nova

- Manages lifecycle of compute instances
- Span, schedule, decommission machines on demand
- Fabric controller – manage and automate pools of resources
- Works with KVM, Vmware, Xen, Hyper-V
- Also bare metal HPC
- Written in Python

- Swift

- Scalable redundant storage system
- Files written to multiple disk drives – replication
- In case of a drive failure, contents replicated to new location
- Analogous to Amazon's S3
- Can store billions of objects across nodes

- Glance

- Image service (Image – OS installed on a VM)
- Manages/stores VM images
- Provides registration services for disk and service images
- Also used to store and catalog backups
- API a standard REST interface for querying info about disk images
- Can use local file system, OpenStack Object Store, S3

Other components

- Dashboard (horizon) – GUI
- Identity service (Keystone) – central directory of users, access control
- Networking (Neutron) – system for managing networks and IP addresses
- Block Storage (Cinder) – persistent block-level storage devices for compute instances
- Database (Trove) – relational and non-relational
- Bare Metal (Ironic) – instead of provisioning VMs

Other PaaS Examples

- AppScale <https://www.appscale.com/>
 - Open-Source framework
 - Supports Google App Engine Applications
 - Supports MySQL Clusters, HBase, Hypertable, and Apache Cassandra
 - Python, Go, and Java applications
- Cloud Foundry <https://www.cloudfoundry.org/>
- open-source PaaS platform that supports programming languages Java and Scala

Other PaaS Examples

- AppFog <https://www.ctl.io/appfog/>
 - Supports Java, Node, .Net, Ruby, PHP, MySQL, Mongo, PostgreSQL
 - Choose your own Cloud Provider.
- Heroku: Cloud Application Platform <https://www.heroku.com/>
 - Was one of the first PaaS services offered on the market
 - Supports Ruby, Java, Scala, and Python applications