Amazon Elastic Compute Cloud (EC2)

Compute

Content Prepared By: Chandra Lingam, Cloud Wave LLC

amazon web services

Amazon Elastic Compute Cloud (EC2)

EC2 is a service to launch virtual server instances

Obtain and boot new instances in minutes

Complete control of the instances

Shutdown or terminate instances when not needed

Amazon Elastic Compute Cloud (EC2)

Multi-tenant Infrastructure

Instance Families

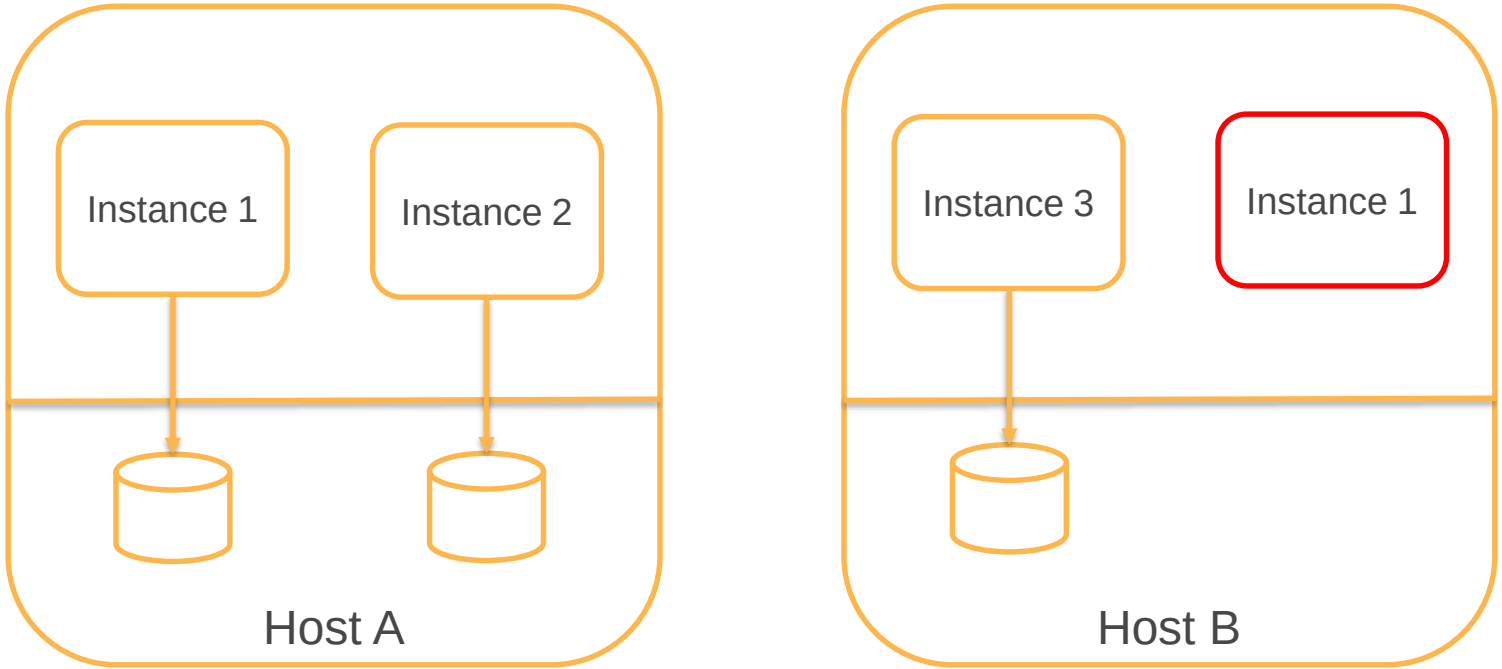
Purchasing Options

Customization

Host and Guest

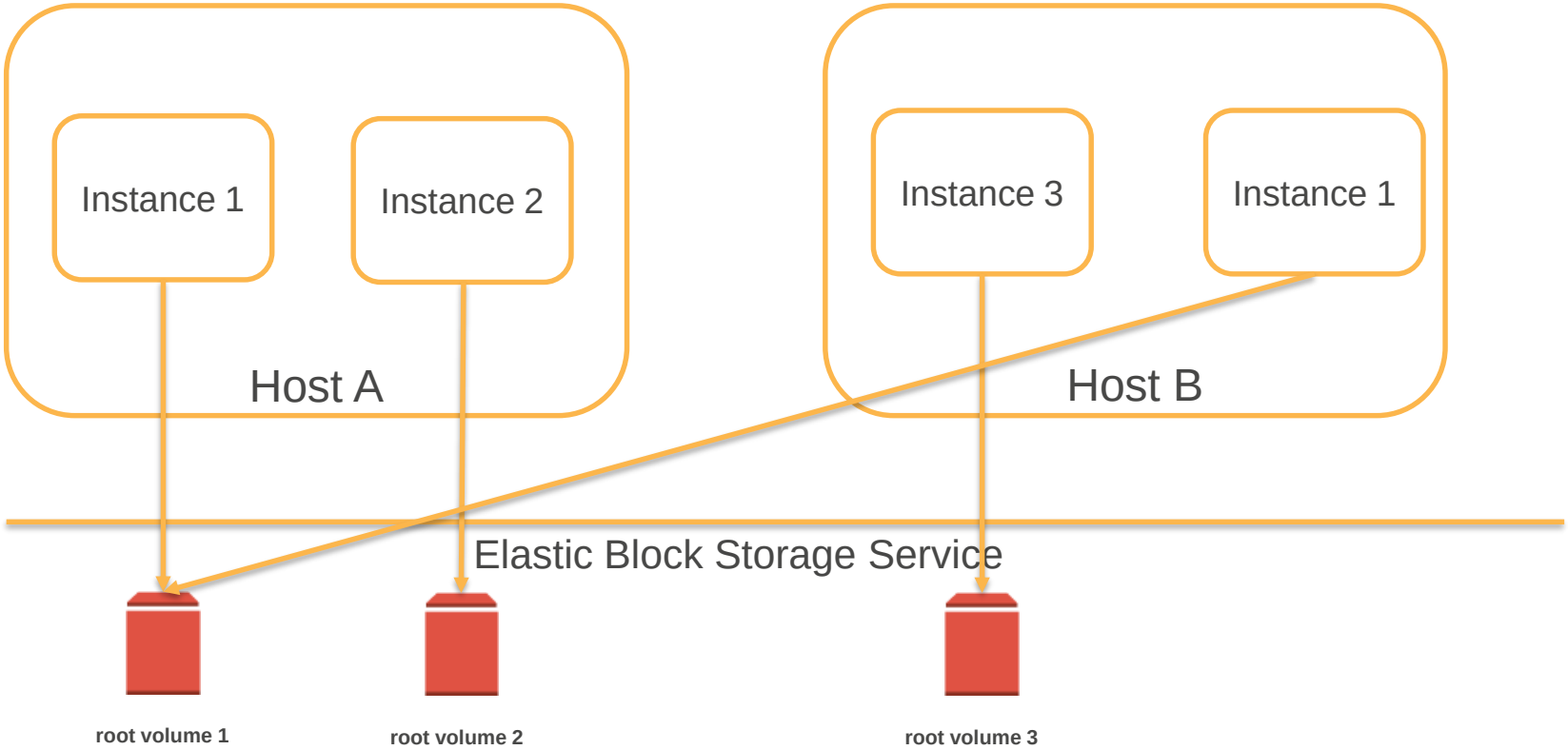
Instances migrate from one host to another when you stop and restart

Storage – Instance Store



Instance store is short lived – Instances cannot be stopped & restarted

Storage – Elastic Block Store



Elastic Block Store is a persistent storage – Instances can stop and restart

amazon web services

Dedicated and Shared Resource

- EC2 dedicates some resources of host computer to each instance: CPU, memory, instance storage
- EC2 shares common resources like disk sub system and network
- When shared resource is underutilized - Instances can consume higher share
- When shared resource are in demand - Instances run at baseline performance

EC2 Hypervisors

Xen Hypervisor

- Amazon uses a customized Xen Hypervisor
- Physical host resources are used by hypervisor

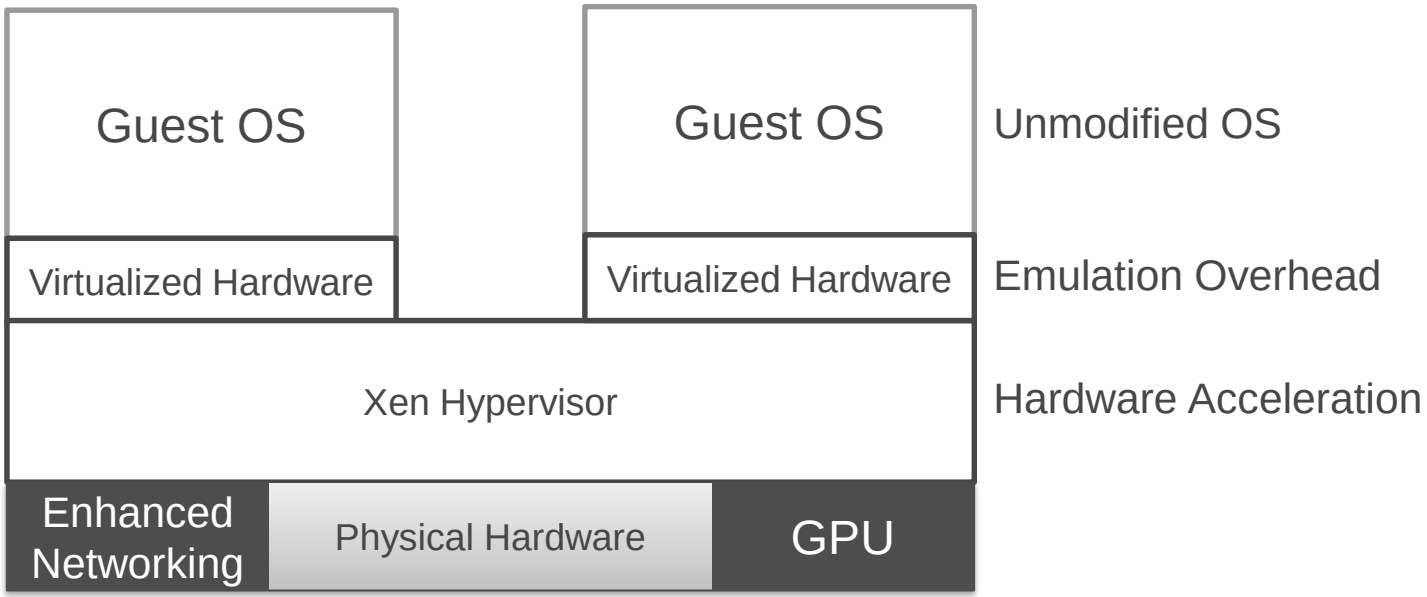
Nitro

- Amazon’s Custom Hardware assisted virtualization
- Light weight virtualization software (derived from Linux KVM)
- Almost all resources of Physical Host available for Guest

Virtualization Types

- Linux OS, choice of virtualization
  - Hardware Virtual Machine (HVM)
  - ParaVirtual (PV)
- Windows OS
  - Hardware Virtual Machine (HVM)

Hardware Virtual Machine (HVM) Virtualization

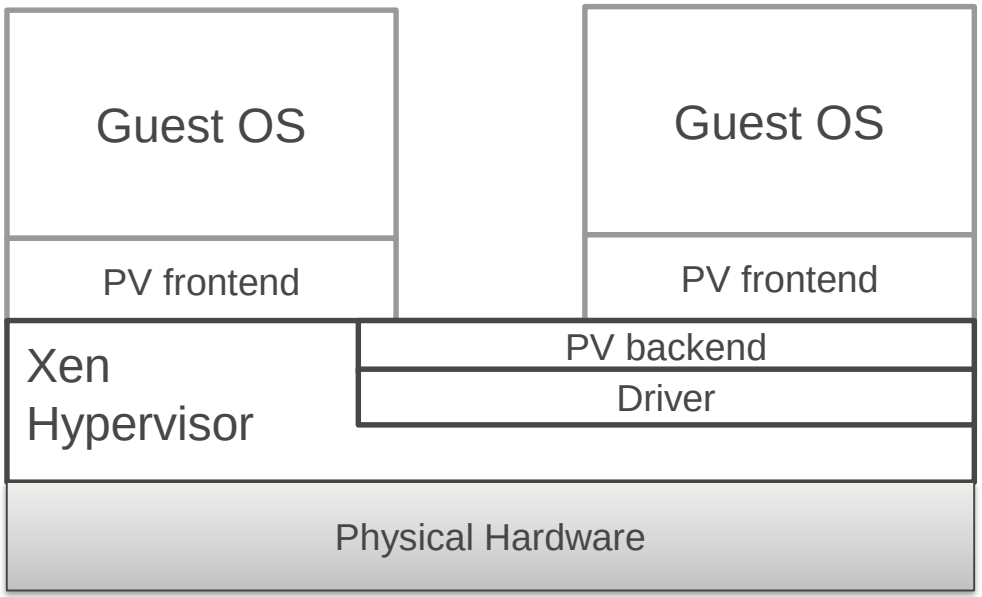


Guest OS can use GPU and Enhanced Networking using Hardware Acceleration

All current generation EC2 instances support HVM

amazon web services

ParaVirtual (PV) Virtualization



Modified OS Kernel

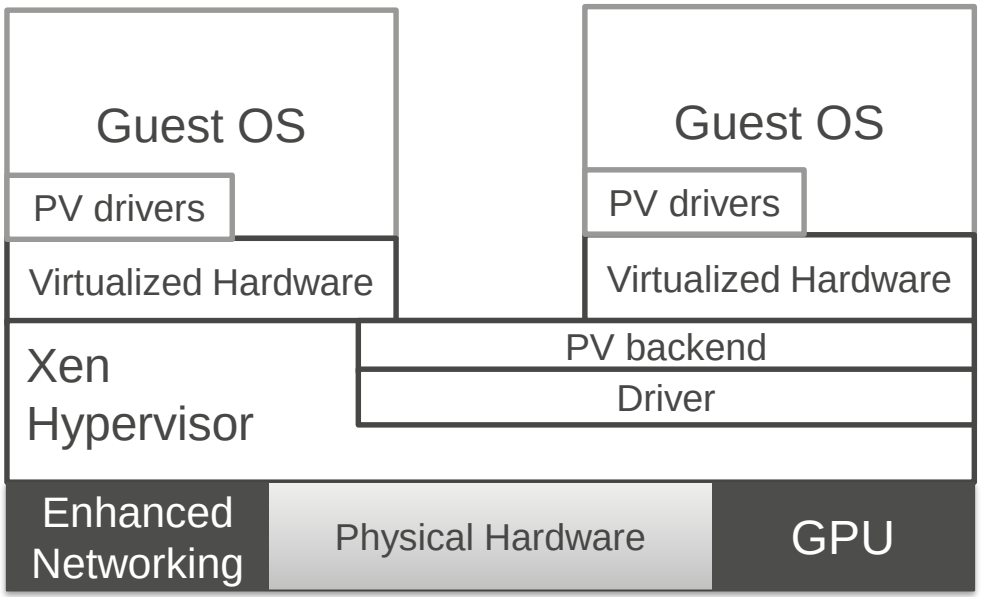
I/O near native performance

Guest OS CANNOT use GPU or Enhanced Networking

PV is phased out for new generation instances

amazon web services

PV on HVM Virtualization

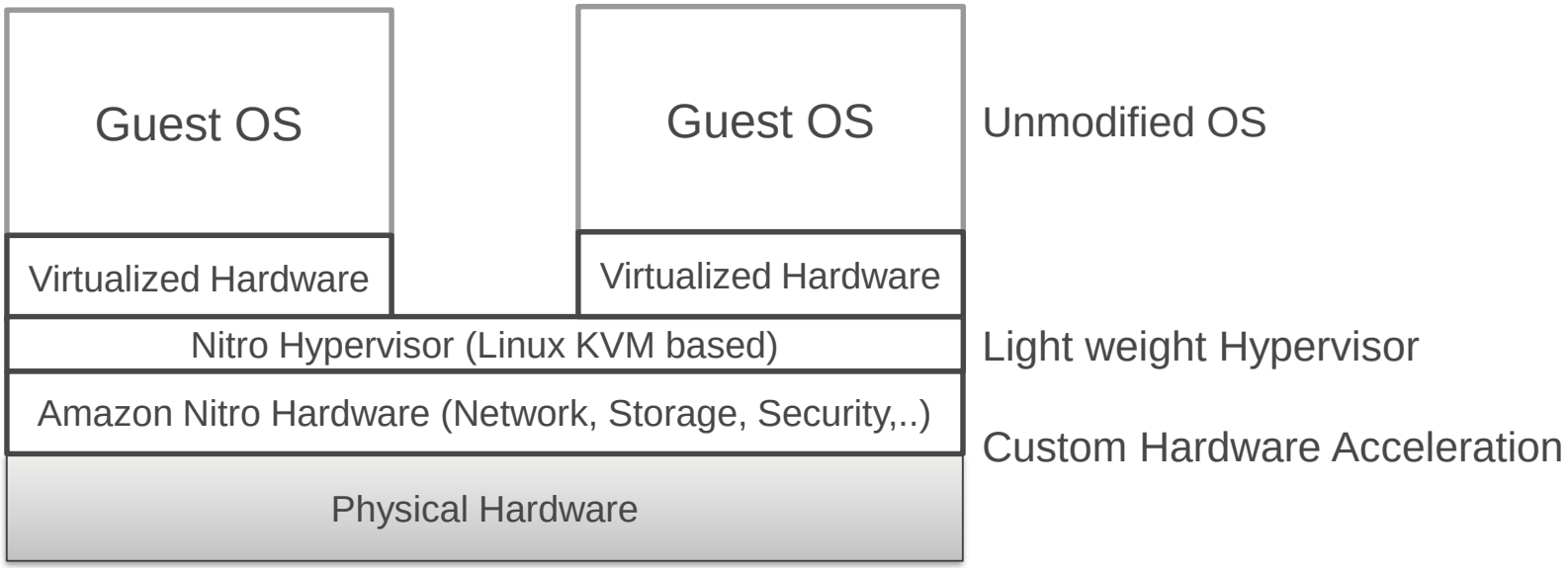


Guest OS can use GPU and Enhanced Networking using Hardware Acceleration

Amazon recommends HVM

amazon web services

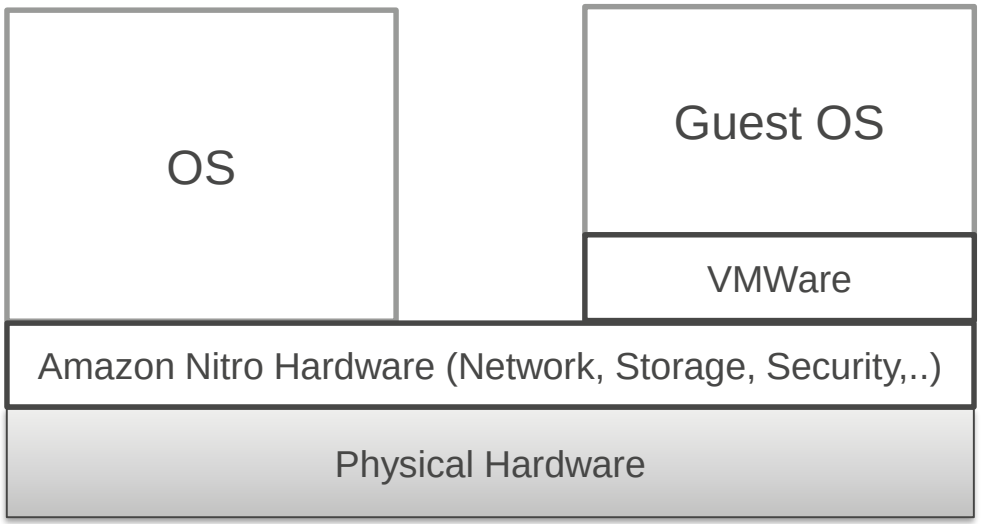
Nitro Virtualization (2017)



AWS is using Nitro hardware acceleration across EC2 instances

- Native hardware performance
- All physical host resources are made available to customers
- *HVM Virtualization Type*

Nitro Bare Metal System (2017)



VMWare on AWS

Virtualized by Nitro Hardware

Bare metal EC2 instances

- Native hardware performance
- Use a different hypervisor like VMware or
- Run your OS directly without virtualization

HVM Virtualization Type

- Runs on bare-metal hardware – from Guest OS perspective
- EC2 host system virtualization layer emulates some or all underlying hardware - using Hardware Assist
- Run Guest OS without any modifications
- Guest OS can access underlying hardware for acceleration.
- Required for GPU processing and enhanced networking
- All current generation EC2 instances support HVM

PV Virtualization Type

- Guest OS aware that it is running on virtualized environment – requires Guest OS Kernel modification
- Delivers higher performance without overhead of system emulation
- Storage and Network I/O see near native performance
- Cannot take advantage of hardware extensions – GPU or enhanced networking
- Some of the current generation EC2 instances support PV

PV on HVM

- PV drivers traditionally performed better than HVM for storage and network – avoids overhead of emulation
- PV drivers are now available for HVM guests
- OS that cannot be ported to PV (Windows) can use PV drivers to match the performance of PV.
- With PV on HVM drivers, HVM guests can get the same or better performance than PV guests

For Best Performance, Amazon recommends HVM with current generation instances

Operating Systems

- Numerous Linux distributions
  - Amazon Linux, Red Hat, SUSE, Fedora, Ubuntu and more
- Microsoft Windows
- FreeBSD - marketplace

Amazon Machine Image (AMI)

- Amazon Machine Image provides information to launch an instance
- Template for root volume: OS, application server, applications
- Additional volumes that needs to be attached to the instance
- Permissions on who can launch an instance
- Several choices from Amazon, vendors and community
- Create your own, buy, share, and sell

Amazon Linux AMI

- Amazon provided and maintained Linux image
- Stable, secure, high-performance environment for EC2
- No additional charge
- Repository access to multiple versions of common packages
- Updated on regular basis include latest components
    - Can be used to update running instances through repository
- Includes AWS packages for integration – CLI, API, AMI tools, Boto library for python, ELB tools

Instance Families

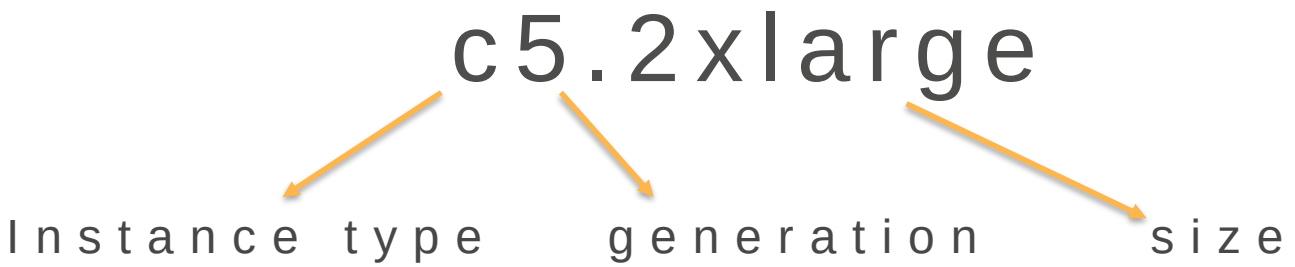
Choice of CPU, Memory, Storage, Network, Graphics, Hardware Acceleration for your needs.

Determines the hardware of the host computer used

EC2 Instance Family

Instance Family	Strength/Uses
General Purpose	Balanced performance suitable for many business applications
Compute Optimized	CPU intensive workload
Storage Optimized	Very high random I/O and sequential I/O with local storage
Memory Optimized	In-memory databases, cache and analytics
Accelerated Computing	Graphics Acceleration, Graphics Compute, Fractional GPUs, Custom Hardware Acceleration
Bare metal	Direct access to underlying hardware. Full access to all AWS services

Instance Type and Size



c5.2xlarge = Compute Optimized, 5th generation, 2xlarge (8 vCPUs, 16 GB Memory)

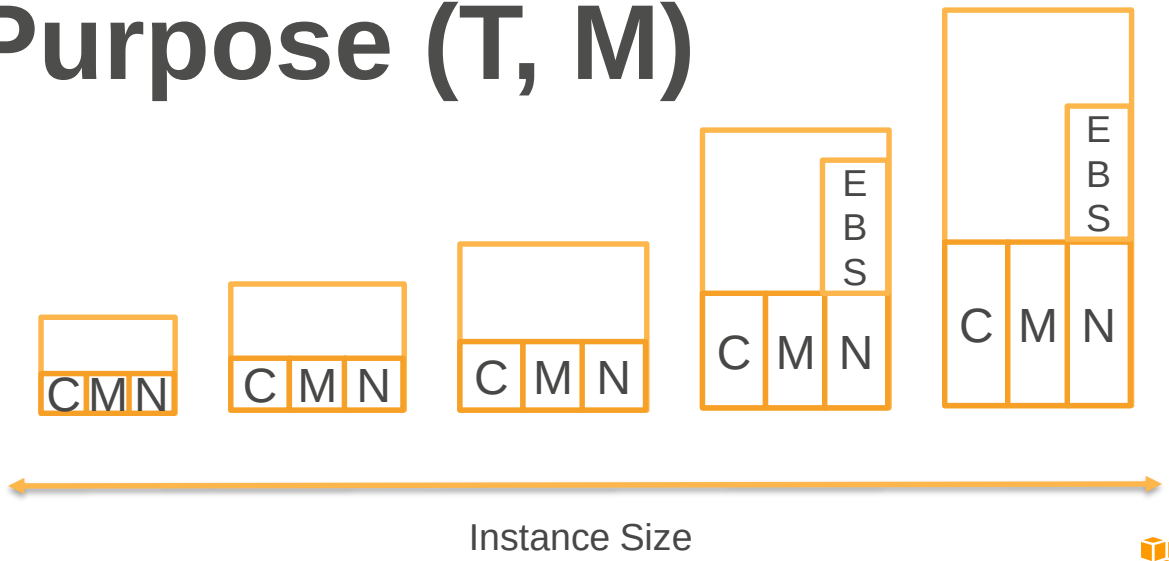
Hardware Acceleration

EBS Optimization - Provides consistent throughput and I/O performance using dedicated bandwidth for EBS

Enhanced Networking – Higher Bandwidth, Higher Packets per second, Lower Jitter

NVMe – Low Latency, High Performance Interface for SSD Instance Storage. AWS also supports NVMe for EBS

General Purpose (T, M)



General Purpose

Instance Type	Usage
Fixed Performance (M4,M5)	Applications that have consistently high CPU utilization

Uses: Web/Application Servers, Small/Medium Databases, Gaming Servers, Caching Fleets and other enterprise applications

Balanced performance suitable for many business applications

General Purpose

Instance Type	Usage
Burstable (T2,T3) Standard Mode	Most business applications rarely use high CPU all the time T2/T3 provides full access to very fast CPUs when needed Continuously accumulates CPU Credits and uses it for higher performance Throttles CPU to baseline when out of CPU credits Launch Credit (30 per vCPU) provides good startup experience

Uses: Virtual Desktops, Microservices, Dev/Build/Stage Environments, Code Repositories, Proof-of-concept systems, Small/Medium Databases

CPU Credits Explained

T2/T3 Hardware Specification

CPU Credits and Baseline Performance

One CPU Credit
1 vCPU running at 100% utilization for 1 minute
1 vCPUs running at 50% utilization for 2 minutes
2 vCPUs running at 50% utilization for 1 minute and so forth

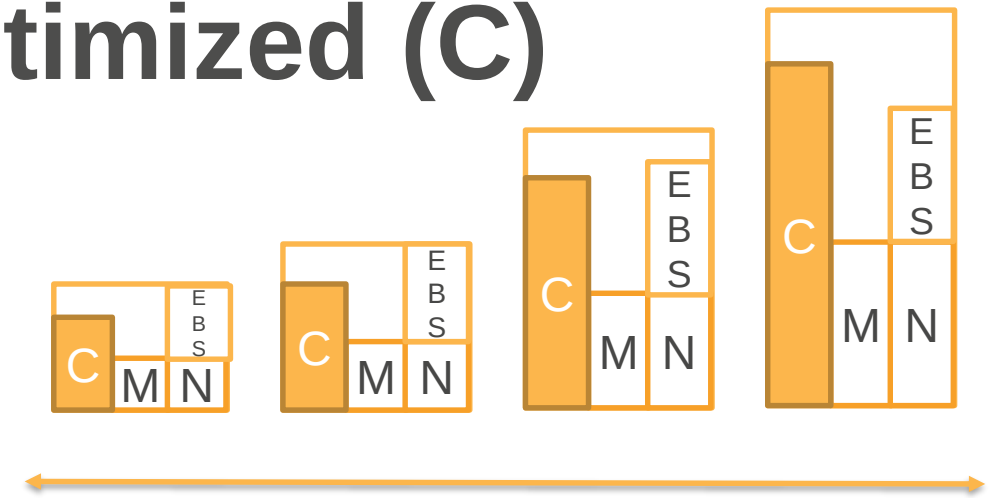
A short burst of CPU uses a small fraction of CPU credit (at millisecond resolution)

CPUCreditUsage – Number of CPU Credits spent by the instance
CPUCreditBalance – Number of CPU Credits accrued by the instance

General Purpose

Instance Type	Usage
Unlimited Mode (T2,T3)	Burstable instance type throttles performance to baseline when out of CPU Credits For critical applications, throttling may not be desirable. Unlimited mode allows instance to sustain high CPU performance for any period of time Pricing similar to burstable instance when average CPU over 24 hour window is at or below baseline Pay Additional vCPU hourly charge when higher performance is needed (average over 24 hour window is above baseline)

Compute Optimized (C)

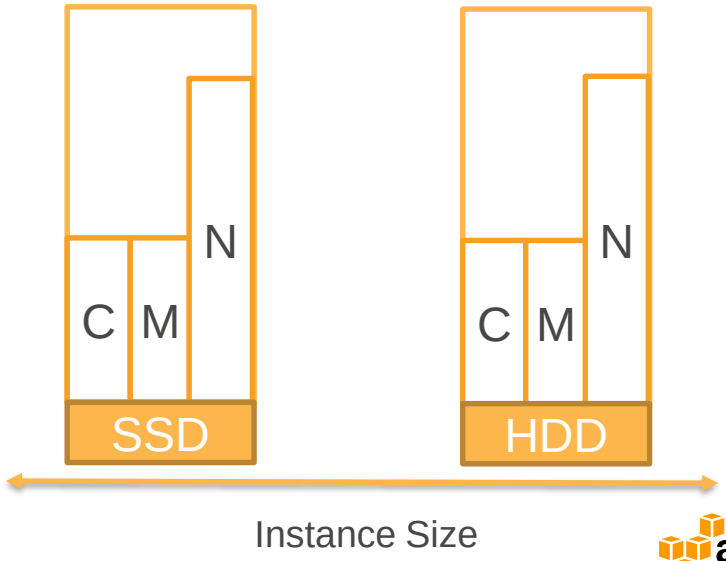


Compute Optimized

Instance Type	Usage
C4,C5	CPU intensive workload High Performance Latest Generation Processors

Uses: Batch processing, Media Transcoding, High performance webservers, High performance computing, Gaming Servers, Ad Engines, Machine Learning

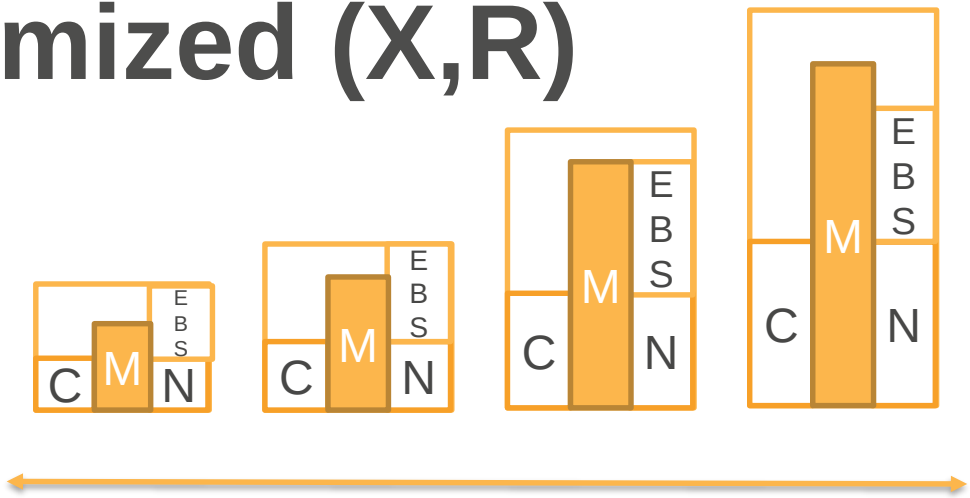
Storage Optimized (I,D,H)



Storage Optimized

Instance Type	Usage
SSD Instance Store (I2,I3)	Very High Random I/O (over 3 Million IOPS) Low Latency Local Data Access Uses: High frequency OLTP Systems, NoSQL Databases, Cache, Low Latency Ad-Tech
Magnetic Instance Store (D2,H1)	High Density (H1 - 16 TB, D2 - 48 TB) High Sequential I/O (throughput up to 3.9 GB/s) Suitable for large data set sequential access Uses: MapReduce, Hadoop Distributed File System, Log and Data Processing Applications

Memory Optimized (X,R)

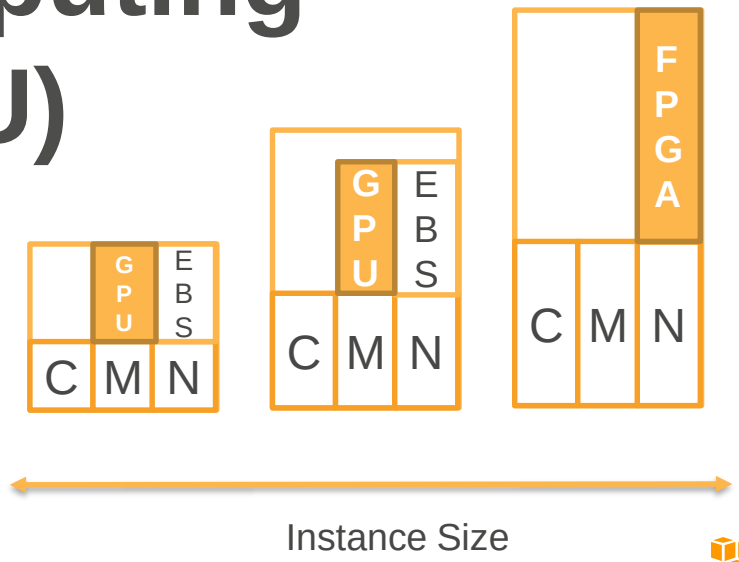


Memory Optimized

Instance Type	Usage
R4 (8:1 memory to CPU)	High performance databases
X1 (16:1 memory to CPU)	In-memory caching
X1e (32:1 memory to CPU)	In-memory analytics
	Real-time processing

Uses: Relational Databases, NoSQL, Redis, Memcached, SAP HANA, Real-time processing of big unstructured data

Accelerated Computing (P,G,F,Elastic GPU)



Accelerated Computing

Instance Type	Usage
Elastic GPU	Low cost fractional GPU capacity that you can attach to any supported EC2 instance
Graphics Intensive (G2,G3)	High performance platform for DirectX, OpenGL graphics applications Uses: 3D rendering, video encoding, virtual reality, graphics-intensive remote desktops
GPU Compute (P2,P3)	General Purpose GPU Computing using CUDA or OpenCL Uses: Deep Learning, Graph databases, computational fluid dynamics, seismic analysis
FPGA (F1)	Custom Hardware Acceleration for computationally intensive algorithms

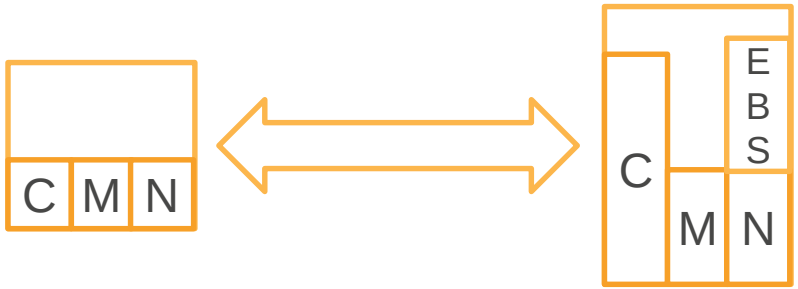
Demo – EC2 Burstable CPU Credits

- Burstable Instance Performance is governed by available CPU Credits, Instance Type and Size
- Observe how CPU Credit influences performance on t2.micro instance in Standard Mode
- Observe how Unlimited mode enables higher performance during unexpected heavy traffic
- Sysbench Tool for CPU Performance Benchmarking

Demo – EC2 Webserver

- Launch EC2 Instance
- Install Apache Webserver
- Configure Test Page
- Take it through Reboot, Stop and Start Cycles
- Elastic IP

Resizing Instances



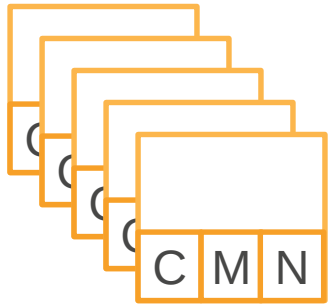
Resizing Instances

- Resize an existing instance based on your usage – over or under utilization
- Stop instance, Change instance type/size, Restart
- Needs to have EBS root device volume
- Target instance type must be compatible
    - Virtualization Type. HVM <-> PV not allowed
    - 32 bit <-> 64 bit not allowed
    - Some instances are restricted to VPC. You cannot use in EC2-Classic
    - Enhanced Networking, NVMe

Demo – Resize - Sysbench CPU Benchmark

CPU	t2.large		t2.micro	
	Single Thread	Two Threads	Single Thread	Two Threads
Events Per Second	912	1,822	857	860

Placement Group



Placement Group - Network for High Performance

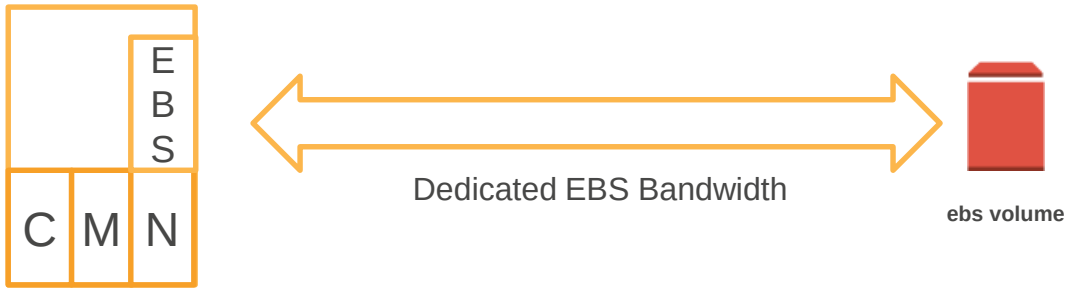
- Logical grouping of instance in a single AZ for High Performance Computing
- 10Gbps for single flow traffic, up to 25 Gbps aggregate flow (provided instance type supports 10 Gbps or more)
- Use Enhanced networking – Higher packets per second, lower network jitter, and lower latency
- There is no extra charge
- Recommended - launch all instances you need together and use single instance type

NEW: This is now referred as Cluster Placement Group

Placement Group – Spread (NEW: 2018)

- Cluster placement group can share same underlying hardware to maximize network performance
- Spread placement group minimizes risk of simultaneous failures due to underlying common hardware
- Spread provides access to distinct hardware and supports mixed instance types, added gradually
- Multiple Availability Zones are supported
- Region: Supports 5 Gbps for single flow traffic, up to 25 Gbps for multi-flow

EBS Optimized Instances



EBS Optimized Instances

- Storage for High Performance
- EBS optimized instances – provides additional, dedicated capacity for EBS I/O
- Minimizes contention between EBS I/O and other traffic from the instance
- Throughput ranges from 500 Mbps to 10,000 Mbps based on instance types
- With EBS Optimized Instances, Provisioned IOPS volumes deliver within 10% of provisioned IOPS 99.9% of the time

Secure Login Key Pairs

- EC2 uses public key cryptography for login
- When you launch the instance, you need to provide the public key
- Maintain the private key (remains with you) in a secure location
- You need the private key to logon to the system using SSH. Linux instances do not have a password
- For Windows instances, you need private key to decrypt the administrator password and then logon using RDP
- EC2 uses 2048 bit SSH-2 RSA keys

Physical Location

- Pick a location where you want to launch your instance
- Launch instances in multiple availability zones in a region for fault tolerance
- Launch instances across multiple regions for disaster recovery, compliance, improve latency by keeping it closer to customer

Demo – Windows Instance

- Launch Windows Instance
- Keypair for admin access
- Configure Security Groups
- Extract password using keypair
- Connect to instance with RDP
- Terminate

Network

- Launch instances in your virtual private cloud (VPC)
  - Assign your own address range
- Keep instances in public subnet – for internet accessible systems
- Keep instances in private subnet – to restrict access and reduce footprint

Bastion Host

- Bastion Host is used to access your private resources from public internet
  - EC2 instances in private subnet allows SSH/RDP only from Bastion Host
  - Bastion Host on public subnet – allows access from specific IP address range for SSH/RDP access
- Reduce attack surface by controlling access points
- Harden to protect your resources
- Do not place your private key in bastion host – use SSH agent forwarding for connecting to private EC2 instances
- Windows Remote Desktop Gateway

Demo – Bastion Host

- Launch a VPC with public, private subnet with NAT device
- Launch an instance in private subnet with security group AppServerSG
- Launch a bastion host in public subnet with BastionSG security group
- Putty - Configure SSH client for SSH Agent Forwarding
- Connect to bastion host
- Connect to AppServer instance from bastion host
- Terminate all instances

Network and Security

Security Group

- Stateful – If a request is allowed, its response is also allowed irrespective of what the response rule says.
    - If instance is allowed to receive a request, it can also respond to the request irrespective of the outbound rules
    - If instance is allowed to send a request, it can also receive the response irrespective of the inbound rules
- Mandatory firewall for EC2 instances – applies to all inbound, outbound traffic at instance level
- Security Group enforced at Hypervisor layer

Security Group

- All rules are evaluated by AWS whether to allow the traffic
- Security Group to Instance association:
  - A security group can be attached to many instances
  - An instance can have many security groups
- You can only specify what traffic is allowed; cannot add deny rules
- Modify rules any time – new rules are automatically applied to all instances

Security Group

- Due to Stateful nature, some communication may be allowed on existing connections even if the new rules are different
    - Use network ACL if you want to immediately stop the traffic
- You can identity source or target using IP Address, CIDR Block, or by specifying another security group.
- If security group is specified as source or destination, all instances belonging to that group get the access.

Security Group

- You can specify security group belonging to a peer VPC connection
- If peer VPC connection or peer security group is deleted, then entry is marked stale – you need to manually remove the entry

Security Groups – EC2 Classic

- EC2-Classic security group needs to be assigned to an instance at launch. Cannot assign a different security group later
- Security Groups in EC2-Classic are at region level
- You can add or remove rules
- You can attach up to 500 security groups with up to 100 rules in each security group
- EC2-Classic and EC2-VPC have their own security groups

Security Groups – EC2 VPC

- In EC2-VPC, Security groups are at VPC level
- Attach different security groups after launch
- Security Group is attached to network interface
- Instance security group is really attached to primary network interface *eth0*
- IPv6 requires separate set of rules

Default Security Groups

- AWS provides a default security group per VPC and per region in EC2-Classic
- If you launch an instance without security group, it is attached to the default group
- Default Security Group Rules
    - Inbound – allows all traffic from other instances in the default security group
    - Outbound – allows all traffic

Custom Security Group

- You can create custom security groups depending on the role played by the instance
  - WebServer Security Group,DB Server Security Group
- Custom Group Default rules:
  - Inbound – no traffic allowed
  - Outbound – allows all traffic

Network Access Control Lists (ACL)

- Controls traffic in and out of a subnet
- Default ACL for every VPC - All inbound traffic and outbound traffic allowed
- Custom ACL - All inbound traffic and outbound traffic denied
- Subnet can have one ACL – can be replaced with another ACL
- One ACL can be attached to multiple subnets

Network Access Control Lists (ACL)

- Numbered list of rules and evaluated in increasing order
- Each rule can allow or deny traffic
- Rules are stateless
    - Inbound requests are subject to inbound ACL rules and corresponding response is subject to outbound ACL rules
    - Outbound requests are subject to outbound ACL rules and corresponding response is subject to inbound ACL rules
    - Can instantly block traffic if needed
- Default Deny rule with rule number *

Controlling Access

- Use IAM to control access to users who can manage EC2 resources
- Use EC2 Instance Login management with Key Pairs for instance OS login access
- Keep your AMIs and EBS snapshots private or share with other accounts (any user in those accounts can access)
- Make AMIs public if you want to share with everyone

IAM Roles

- Grant instance access to other AWS services using IAM roles
- Application running on Instance can get temporary credentials using metadata service
- These temporary credentials have privileges granted through IAM role permissions and can access other AWS services
- No need to maintain Access Key / Secret Access Key pairs
- You can attach an IAM role to a new instance or to an existing instance.

Demo – IAM Roles

- Create EC2 Service IAM Role
- Attach Permissions to Read S3 Bucket
- Attach Role to EC2 instance
- Confirm S3 access from EC2 instance

Instance IP Addressing

- EC2 and VPC support IPv4 and IPv6 addressing
- Default is IPv4 and you cannot disable it. Specify a private IPv4 CIDR block during VPC and subnet creation
- Optionally, you can assign IPv6 CIDR block to VPC and subnet. Not supported in EC2-Classic
- Static Private Address – Each instance in VPC is assigned a static private address that is released only on instance termination
- Secondary Private Address can be assigned to an instance if needed

Instance IP Addressing

- Internal DNS Hostname: Each instance is also given a internal DNS hostname that resolves to private IP address. Example: ip-10-251-50-12.ec2.internal
- Public IP Address and External DNS Hostname – Optionally, instance in VPC can receive a public IP Address and External DNS hostname. This is controlled at VPC/Subnet level. Default VPC grants public ip; non-default VPC grants only private IP
- External DNS Hostname resolved to public IP when queried externally and resolves to private IP when received inside VPC

Elastic IP

- Static Public IPv4 Address
- Limited to 5 IPv4 addresses per region per account – IPv4 public address is a scarce resource (you can request additional if architecture warrants)
- Unused Elastic IP Addresses (including stopped instances) are charged an hourly fee.
- You are not charged for one Elastic IP address associated with the running instance
- You are charged for any additional elastic IP addresses associated with the instance

VPC Instance Network Interface

Network interface attributes list

Source/Destination Checking

- Disabling this attribute enables instance to handle network traffic that isn't specifically destined for this instance
- NAT or routing requires this to be disabled. Default is enabled for all instances.

Instance State and Actions

Instance Lifecycle

Instance States – Pick AMI and Launch

State|Action	Description
Pending	Instance enters pending state once launched. Boot the instance with AMI specified
Running	Instance is ready for use. Billing Starts
Reboot	Use EC2 tools to reboot instead of running OS reboot command
Stop (EBS)	Enters stopping state (Billing stops) and then stop
Start (EBS)	It goes to pending state. Moves to a new host. Each transition from start -> running is charged one billing hour.
Terminate	Instance no longer needed. Billing stops when status is shutting-down or terminated. Volume optionally deleted
Retirement	AWS retires host due to hardware issues. Instance terminated on scheduled retirement date. EBS instances can be restarted on different host. Instance store looses data.

Reboot, Stop and Terminate

Reboot, Stop and Terminate - Differences Table

Recover Instance

- If instance is not reachable, you can automatically recover using CloudWatch Alarm (StatusCheckFailed_System)
  - Loss of network connectivity
  - Loss of system power
  - Software issues on physical host
  - Hardware issues on physical host
- Instance relaunched on a new host and maintains all attributes (public/private/elastic IP/instanceid/all metadata)
- EBS root volume and on shared tenancy (default)

EC2 Instance

Purchasing Options

EC2 Instance Pricing

Two aspects to pricing:

Instance Cost – Based on Instance Type, Size and how instance was purchased

Variable Costs – Based on EBS Storage, Data Transfer, Burstable instance unlimited mode, Elastic IPs

EC2 Purchasing Options

Type	Key Features
On-Demand	Flexible, Uninterrupted No long-term commitment Capacity issues may prevent on-demand launch
Reserved	Long term use of a specific configuration (1 year or 3 year) Substantial discount (up to 75% off the on-demand price) Ideal for continuous usage Optional Capacity Reservation
Scheduled	Usage on a recurring schedule (daily, weekly, monthly) Capacity Reservation Requires 1 year commitment with at least 1200 hours usage 5-10% off the on-demand price
Spot	Steep discount (up to 90% discount off the on-demand price) Bid for unused spare capacity Interrupted anytime – Pause/Resume Option

Single Tenant Physical Host

Type	Key Features
Dedicated Hosts	Bring Your Own Server Bound License (Socket, Cores) Meet Compliance Requirements Per Host Billing (instances on a host are not billed) On-demand or Reserved Pricing Instance Placement Control
Dedicated Instances	May share hardware with other instances from same Account Per Instance Billing (Dedicated Hosts vs Dedicated Instances) On-demand or Reserved or Spot Pricing
Bare Metal Instances	For Specialized workload, Legacy workload, Licensing Restrictions Per Host Billing Bring your own virtualization (VMware for example) On-demand or Reserved or Spot Pricing

Note: Other services like EBS, S3 are multi-tenant

Per Second Billing

- EC2 Billed on per second increments
- One minute minimum
- Applies to on-demand, reserved and spot instances
- Some instances are billed hourly

Reserved Instances

One year or Three year commitment (cannot cancel)

RI Attributes – [Region, Instance Type/Size, Platform, Tenancy, optional AZ]

Price Comparison: AWS Simple Monthly Calculator

Capacity Reservation OR Flexible Discounts

Standard and Convertible RI

Shared across accounts using Consolidated Billing

Reserved Instances – Capacity Reservation

Capacity Reservation - When you specify Availability Zone in your Reservation

- AWS Reserves capacity in that AZ for you
- Discount applies only if you launch an on-demand instance that match the reservation attributes

  [Region, AZ, Instance Type/Size, Platform, Tenancy]

Reserved Instances – Flexible Discounts

Flexible Discounts – When you don't specify availability zone in the reservation

- RI applies to all availability zones in the region
- Discount is applied to instances of any size in the same instance family

Example: 1 x m4.2xlarge Linux RI can cover

1 x m4.2xlarge or 2 x m4.xlarge or 4 x m4.large

Spot Instances

Save Money - discounts up to 90%

Fewer interruptions

- Substantial unused capacity
- New Spot Pricing Model (Launched end of 2017)

Spot Pricing Model

Old:

- Complex Bidding Process
- Unpredictable Spot Price
- More instance interruptions

New (launched towards end of 2017):

- Predictable Spot Price
- Uses long term supply & demand
- Fewer interruptions

Spot Instance Usage

Suitable for

- Containers
- Web servers, API Backends
- Big Data Workload (EMR, Hadoop, Spark)
- Rendering
- Continuous Integration/Continuous Deployment workloads

Spot Instance Best Practices

Spot Instance Pricing Fluctuates based on:

[Availability Zone, Instance type & Size, Platform, Tenancy]

Minimize chance of disruption & Maximize Savings:

- Qualify application to run across multiple instance types and sizes
- Use older generation instance types
- Use multiple Availability Zones
- Ensure application can handle instance interruptions (periodically store data outside, split up the work)
- Use default maximum price (on-demand price)

Spot Terminology

Terminology	Description
Spot Price	Current hourly price of a spot instance Gradually moves based on long-term supply and demand
Spot Pool	Unused EC2 instances of same type, OS, availability zone, and network platform
Maximum Price	Maximum price you are willing to pay for a spot instance EC2 fulfils your request when: Maximum price >= Spot price AND Capacity is available
Spot Instance Interruption	EC2 stops, hibernates, or terminates your instance when: Spot price > Maximum price OR Capacity not available
Spot Fleet	Maintains target capacity (Instances or vCPUs)

Spot Request Types

Type	Description
Request	One Time request for one or more spot instances Interrupted instances are not replaced
Request and Maintain	Persistent request for one or more spot instances Interrupted instances can terminate, stop or hibernate Stopped or Hibernating instance is started again when capacity is available Maintains target capacity
Request for Duration	Request spot instances for 1 to 6 hours (spot block) No interruptions 30 to 50% off the on-demand price

Spot request is valid for the specified time window
Optional On-Demand Portion

Spot Allocation Strategy

Strategy	Description
Lowest Price	Spot instance comes from lowest priced pool. Default strategy
Diversified	Balance spot instances across all pools or specified number of lowest priced instance pool

Spot Integration in AWS Services

Amazon EMR

CloudFormation

Elastic Container Service

Auto Scaling

AWS Batch

…

EC2 Pricing – Variable Costs

AWS Simple Monthly Calculator

Dedicated Host, Dedicated Instance

Elastic IP, Unlimited Mode CPU, Elastic GPU

Data Transfer

Storage

EC2 Data Transfer Pricing

Network Transfer	Pricing
To same availability zone - Private IP	Free
To same availability zone – Public IP/Elastic IP	USD 0.01 per GB
To same region - Public IP/Elastic IP/VPC Peering Connection	USD 0.01 per GB
To another region	USD 0.02 per GB
From Internet	Free
To Internet	USD 0.09 per GB

Intra region between EC2 and some services like S3, DynamoDB and so forth is free

Managing Instance

Managing Software

- EC2 Amazon Linux – two repositories enabled by default (amzn-main and amzn-updates)
- Additional repositories can be added
- yum update to upgrade system and packages

Managing Instance User

- Amazon Linux has a *ec2-user* default account
  - Other OS and AMIs come with their own default account
- You have to specify public key credentials to be attached to the default user when launching instance
- This user is different from IAM users
- You can add new users attaching user's public key to the system

User Data

- Run command at launch using “user data”
- Shell scripts and cloud-init directives
- User data and cloud-init directives only run during first boot cycle when instance is launched
- For more complex scenarios use AWS CloudFormation and AWS Opsworks (Chef)
- User data is not encrypted – so do not put sensitive information
- User Data Log - /var/log/cloud-init-output.log

Instance Metadata

- Query data about instance for managing instance
- Access user data for configuration from instance
- Retrieve instance metadata:
- http://169.254.169.254/latest/meta-data/
- Service is throttled. So, cache frequently needed data

Example query host name, user data:

- `curl http://169.254.169.254/latest/meta-data/local-hostname`
- `curl http://169.254.169.254/latest/user-data`

Categories of instance metadata

Table: Categories of instance metadata

EC2 Systems Manager

- Remote administration
- Automate patch deployment, configuration
- Inventory management
- State management
- For more complex scenarios use AWS CloudFormation and AWS Opsworks

Processor State Control

- C-states control sleep levels that a core can enter when idle
  - C0 = totally awake to C6 = deepest sleep state core is powered off
- P-states control performance levels (frequency)
  - P0 = Highest performance with cores allowed to use Intel Turbo Boost to increase frequency
  - P1 = Maximum baseline frequency
  - P15 = Lowest possible frequency
- Available only on select instance type and instance sizes

Processor State Control

- Default settings optimal for most workloads
- Highest performance with maximum turbo boost frequency – allow cores to sleep to give thermal headroom for other cores
- High performance and low latency by limiting C-states – Tune for Latency versus performance – putting core to sleep/waking it takes time
- Baseline performance with lowest variability by limiting P-states. Consistent performance, use headroom for Advanced Vector instruction

Differences between VPC and EC2-Classic

Table: Differences between VPC and EC2-Classic