Amazon Elastic Compute Cloud (EC2)

Compute

Content Prepared By: Chandra Lingam, Cotton Cola Designs LLC

amazon web services

Amazon Elastic Compute Cloud (EC2)

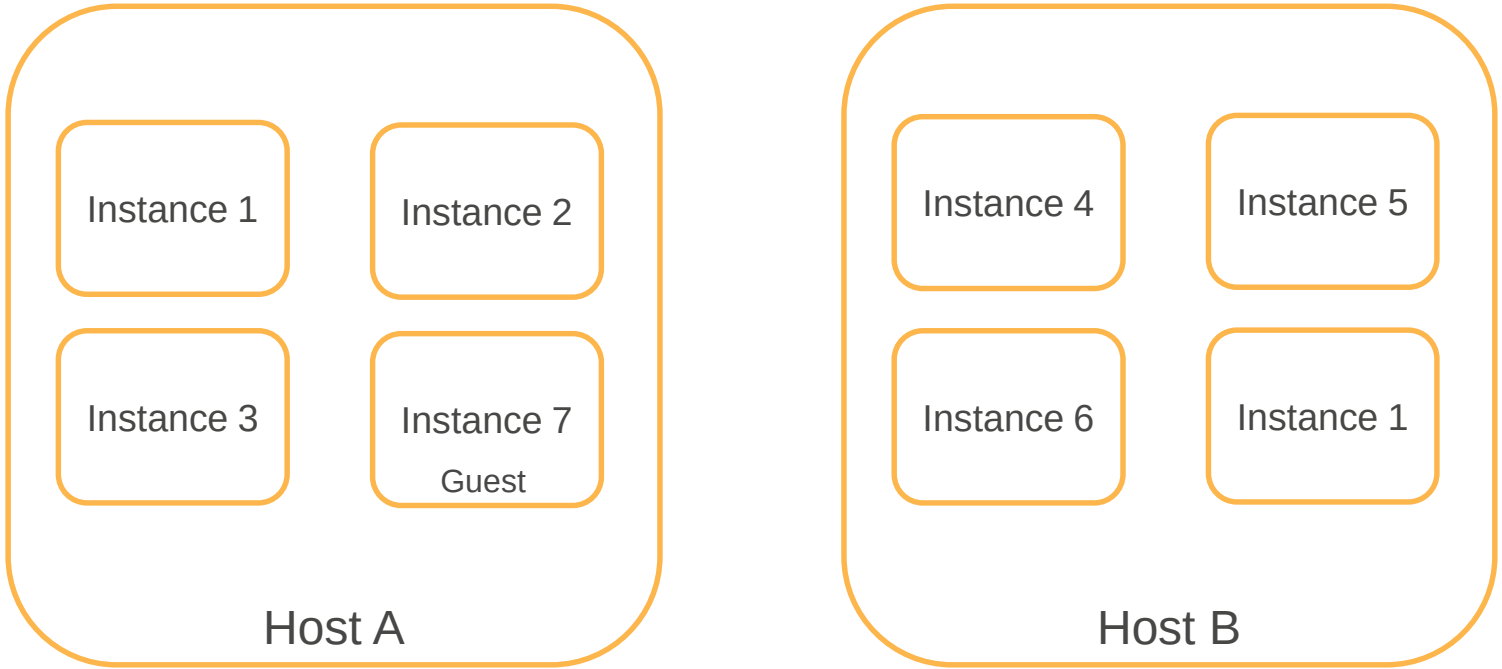
EC2 is a service to launch virtual server instances

Complete control of the instances

Obtain and boot new instances in minutes

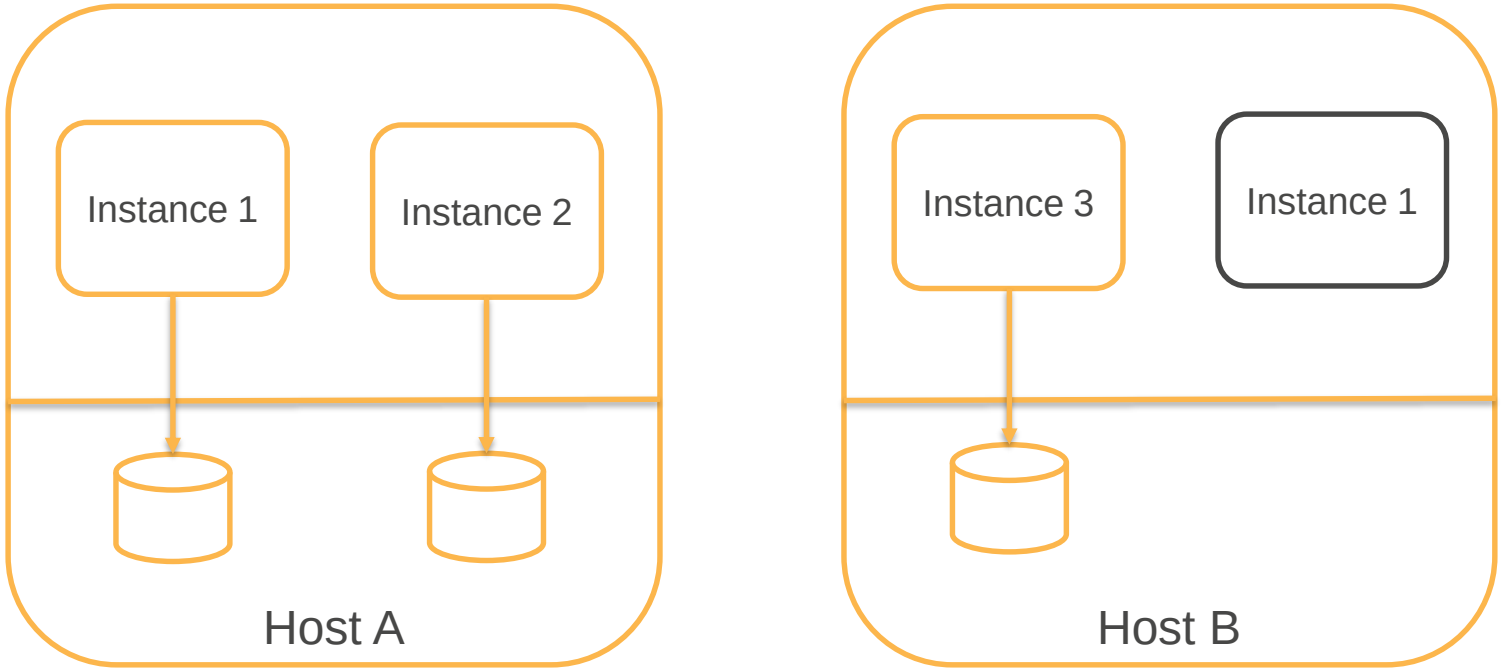
Shutdown or terminate instances when not needed

Host and Guest



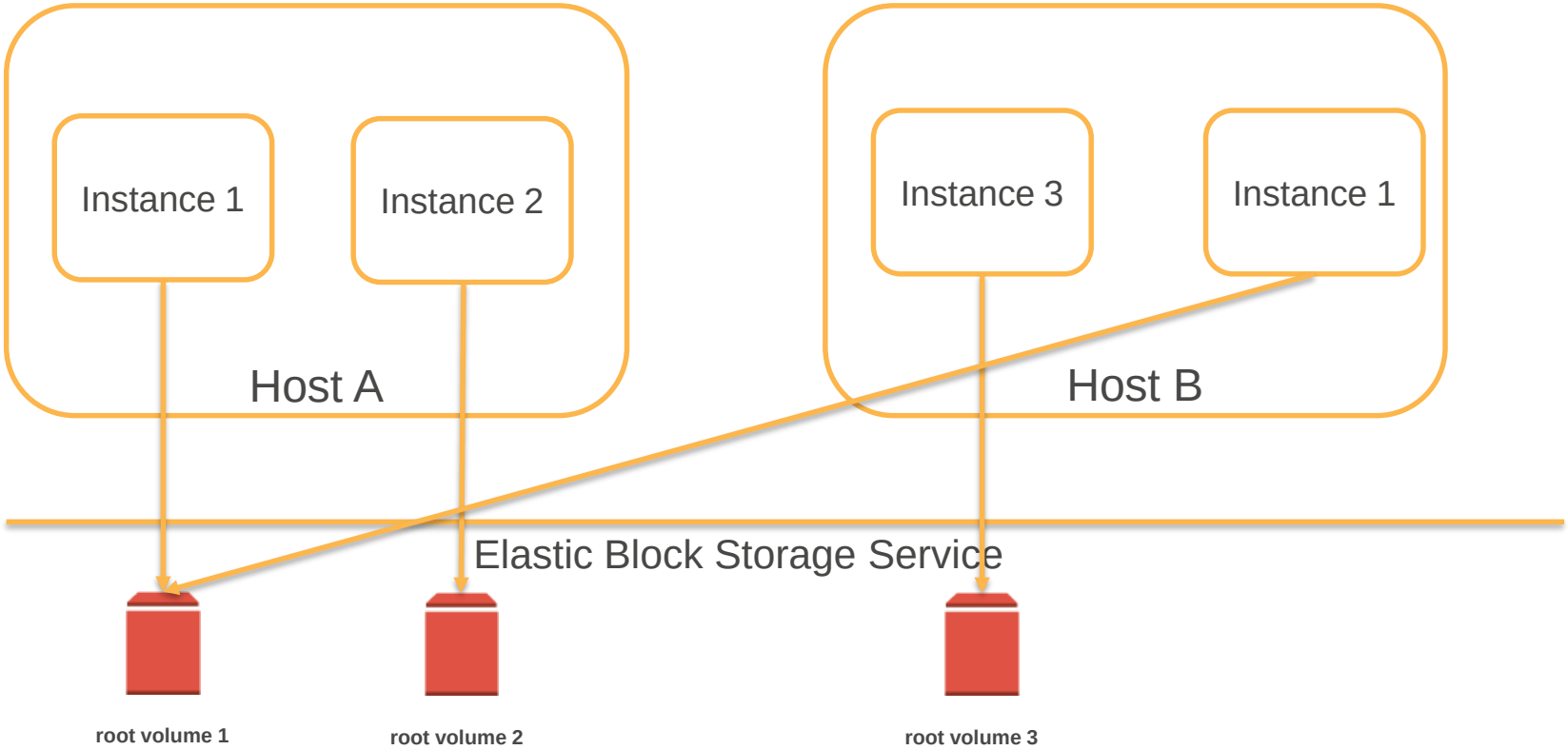
Instances migrate from one host to another when you stop and restart

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 - instance can consumer higher share
- When shared resource are in demand – each receives an equal share
    - High I/O performance instance types allocate larger portion of a shared resource
    - Greater or more consistent I/O performance

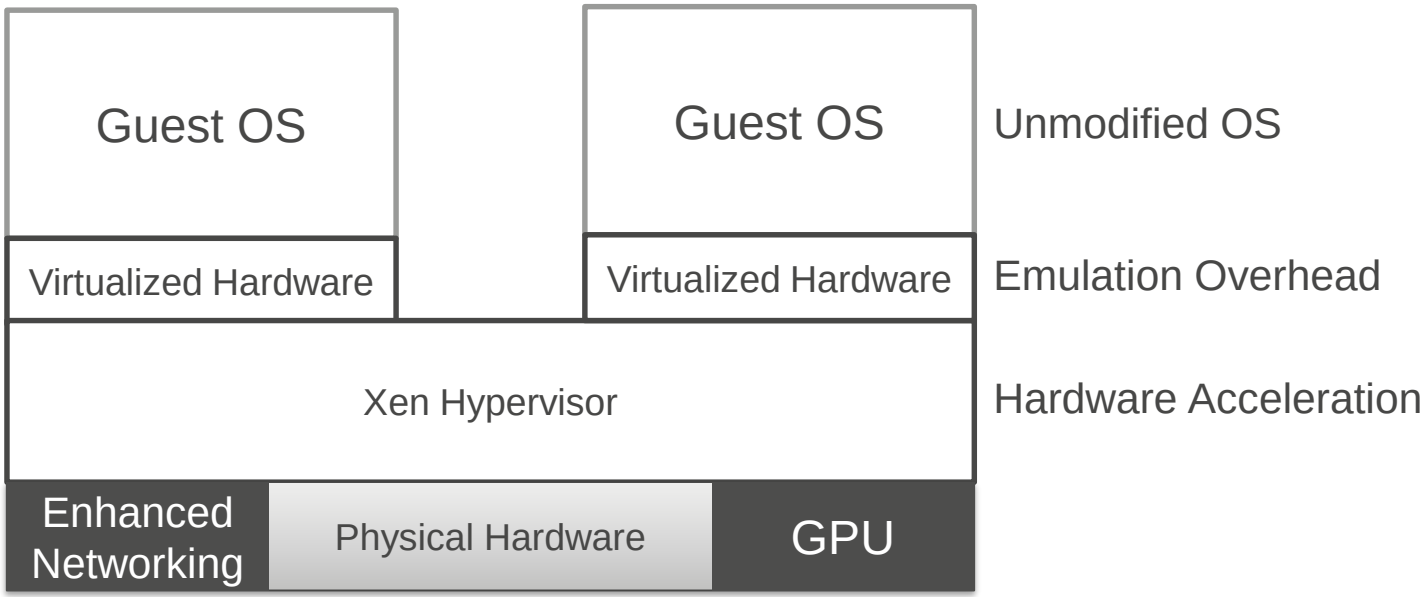
Virtualization Types

- Amazon uses customized version of Xen Hypervisor
- Linux OS, choice of virtualization
  - Hardware Virtual Machine (HVM)
  - ParaVirtual (PV)
- Windows OS
  - Hardware Virtual Machine (HVM)
- Main difference between two virtualization types
  - Boot mechanism, I/O Performance
  - Ability to take advantage of hardware extension (CPU, network, storage)

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

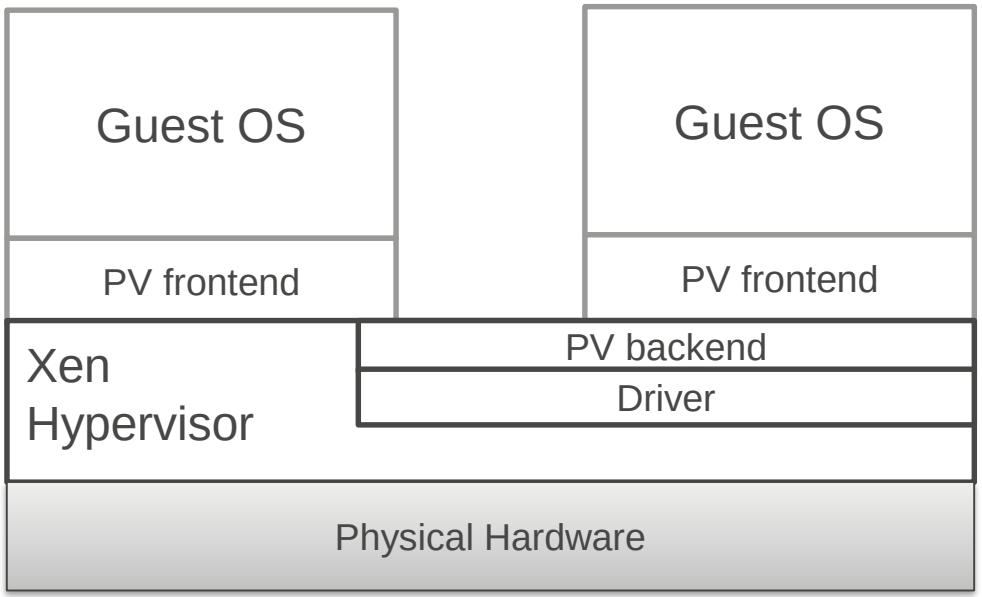
Hardware Virtual Machine (HVM) Virtualization



Guest OS can use GPU and Enhanced Networking using Hardware Acceleration

All current generation EC2 instances support HVM

ParaVirtual (PV) Virtualization



Modified OS Kernel

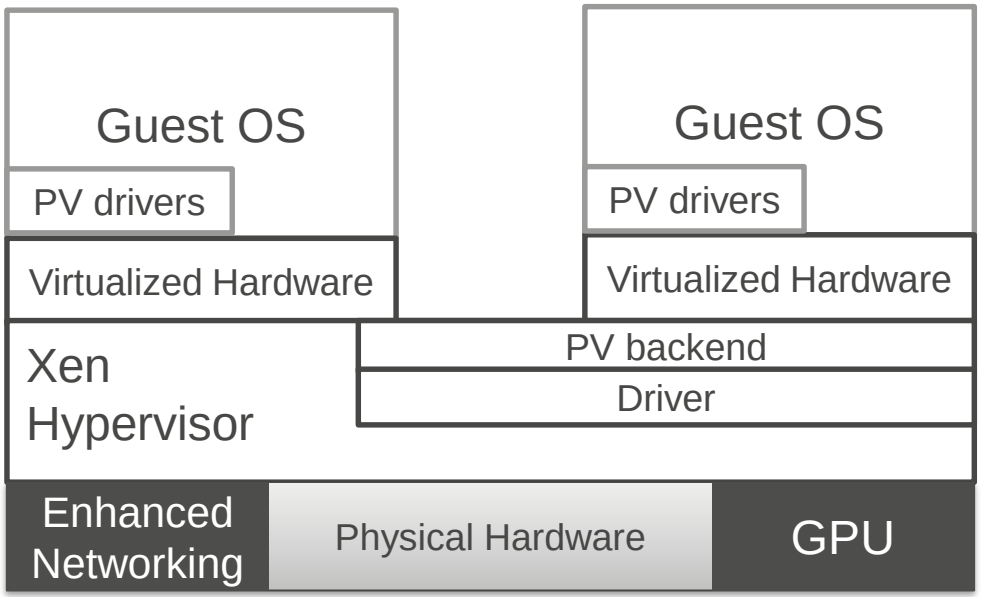
I/O near native performance

Guest OS CANNOT use GPU or Enhanced Networking

Some current generation EC2 instances support PV

amazon web services

PV on HVM Virtualization



Guest OS can use GPU and Enhanced Networking using Hardware Acceleration

Amazon recommends HVM with current generation instances

amazon web services

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

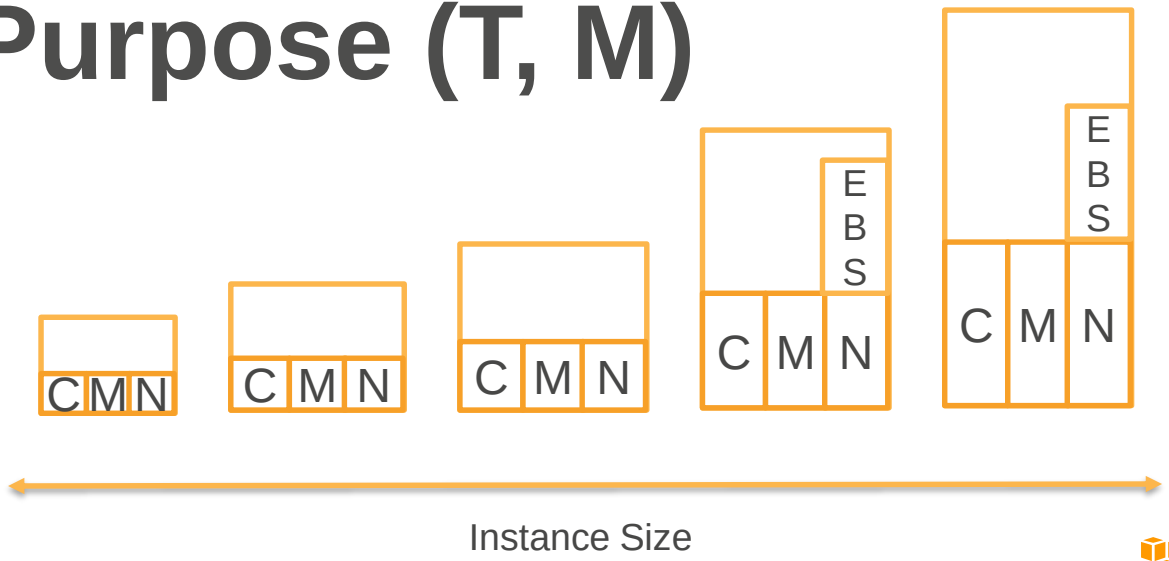
- Amazon provided and maintained Linux image
- Stable, secure, high-performance environment for EC2
- No additional charge
- Repository access to multiple versions of common packages
- Updated on regular basis include latest components
    - Can be used to update running instances through repository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

Instance Families

- General Purpose (T, M)
- Compute Optimized (C)
- Memory Optimized (X, R)
- Storage Optimized (I, D)
- Accelerated Computing (P, G, F)

Choice of CPU, Memory, Storage, Network, Hardware Acceleration for your needs. Determines the hardware of the host computer used

General Purpose (T, M)



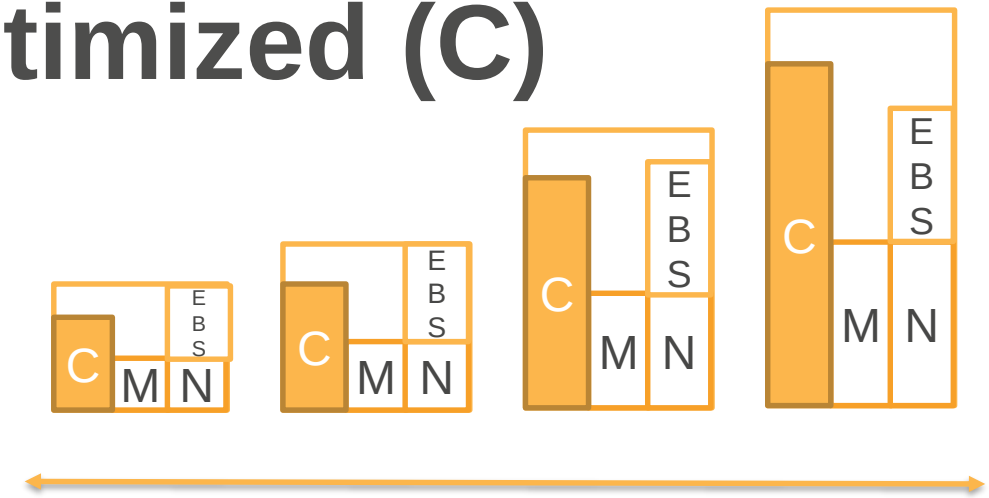
General Purpose – T2 instances

- Lowest cost general purpose instance type - Balance of compute, memory and network resources
- T2. micro eligible for free tier
- Baseline CPU performance with ability to burst
- Burst is governed by CPU credits - Accrue CPU credits when idle and use it when needed
- Good choice for workloads that doesn’t use full CPU but burst occasionally
- Suitable for Webserver, development environments and databases

General Purpose – M4 instances

- Latest generation and provides a balance of compute, memory, network resources
- Good choice for many applications
- EBS optimized at no additional cost
- Support for enhanced networking
- M3 Instance - SSD Based instance storage for fast I/O performance
- Suitable for small-mid sized databases, data processing, cluster compute, sharepoint

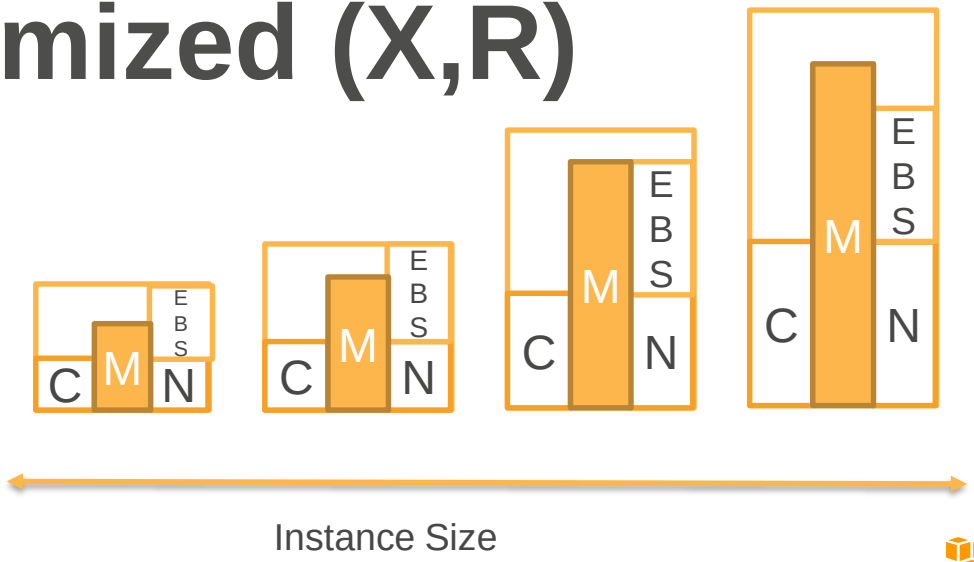
Compute Optimized (C)



Compute Optimized – C4

- Latency gen, highest performing processors
- Lowest price per compute performance in EC2
- EBS optimized at no additional cost
- Support for enhanced networking and clustering
- Ability to control processor C-state and P-state configuration on large instances
- C3 – SSD based instance storage
- MMO gaming, Video encoding, Distributed analytics, batch processing, science and engineering use

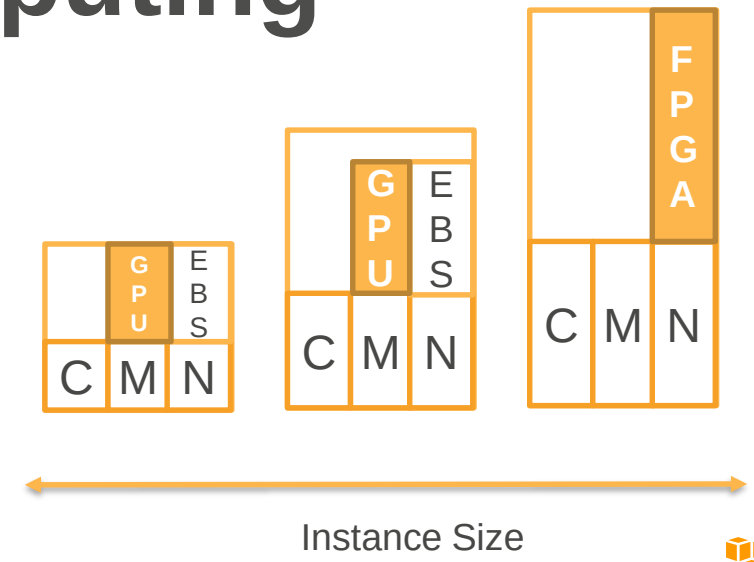
Memory Optimized (X,R)



Memory Optimized – X1

- Optimized for large scale in-memory applications
- Lowest price per GiB of RAM among EC2 instances
- Upto 1,952 GiB of instance memory
- SSD Instance storage
- EBS Optimized at no additional cost
- Ability to control processor C-state and P-state configuration
- Certified -SAP HANA, Apache Spark, Presto, HPC apps
- Smaller R4 and R3 instances available

Accelerated Computing (P,G,F)



Accelerated Computing – P2

- General purpose GPU compute applications
- High performance NVIDIA K80 GPUs
- GPUDirect support for GPU-GPU peer communication
- Enhanced networking upto 20Gbps
- EBS optimized at no additional cost
- Machine Learning, High performance databases, computational fluid dynamics, seismic analysis, rendering, genomics workloads

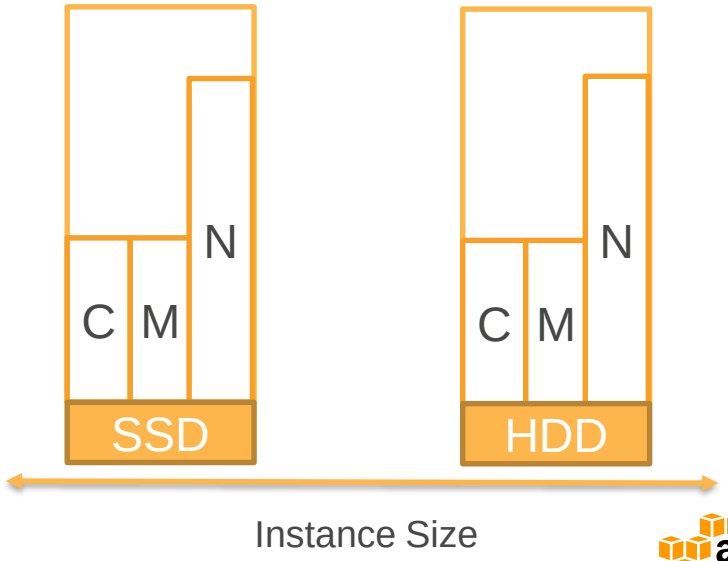
Accelerated Computing – G2

- Optimized for Graphics intensive applications
- High performance NVIDIA GPUs
- On-board hardware decoder for multiple real-time HD streaming
- Low latency frame capture and encoding – High quality interactive streaming experience
- 3D application streaming, video encoding, server side graphic workload

Accelerated Computing – F1

- Customized hardware acceleration with field programmable arrays (FPGA) -Xilinx
- NVMe SSD storage
- Support for Enhanced networking
- Genomics research, financial analytics, real-time video processing, security, big data search and analysis

Storage Optimized (I,D)



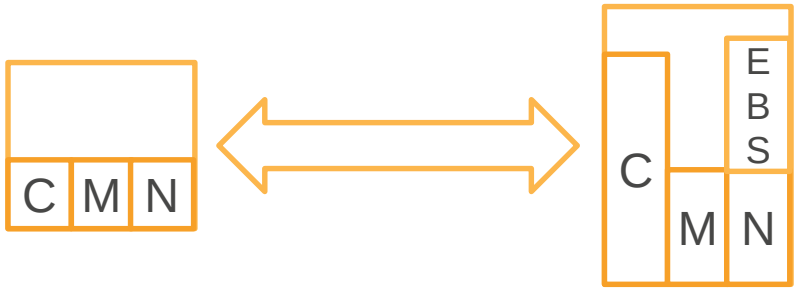
Storage Optimized – I2

- High storage instances with SSD backed instance storage
- Very high random I/O performance
- High IOPS at low cost
- Support for enhanced networking
- NoSQL databases Cassandra, MongoDB, scale out transactional databases, cluster filesystems, data warehousing, hadoop

Storage Optimized – D2

- Dense storage instances – 48TB of HDD local instance storage
- High disk throughput
- Lowest price per disk throughput
- Massively parallel data warehousing, Hadoop Map Reduce, Distributed file systems, network file systems, log or data processing applications

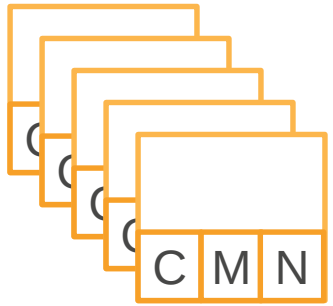
Resizing Instances



Resizing Instances

- Resize an existing instance based on your usage – over or under utilization
- Stop instance, update to new instance type, restart
- Only supported for Instances with EBS root device volume. Not supported on Instance store root device volumes
- 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

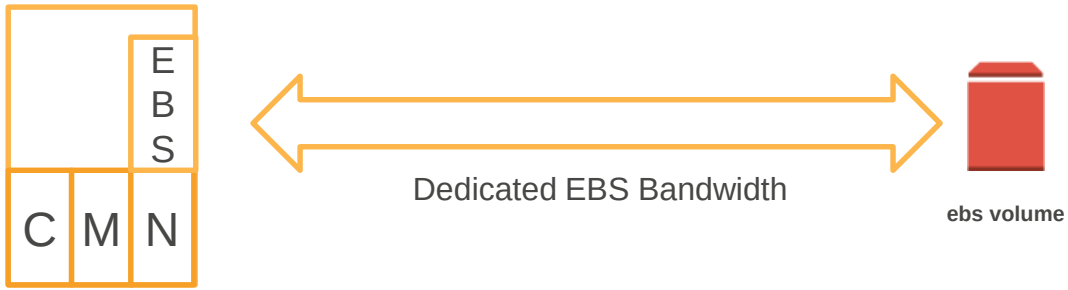
Placement Group



Placement Group - Network for High Performance

- Logical grouping of instance in a single AZ
- Launch supported instance types into a placement group
- Instances in common placement group benefit from 10Gbps, low-latency networking
- Instance types should support 10Gbps
- There is no extra charge
- Recommended - launch all instances you need together
- Use with Enhanced networking – Higher packets per second, lower network jitter, and lower latency

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

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

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

Firewall

- Security Groups – Mandatory firewall for EC2 instances
  - Applies to all Inbound and outbound traffic at Instance level
  - Stateful filters
- Network Access Control Lists (ACL)
  - Applies to all inbound and outbound traffic from a subnet in VPC
  - Stateless traffic filters

Demo – Linux Instance

- How to connect to EC2 instance
- Download Putty
- Generate Key Pairs - Extract Public, Private Keys
- Launch Linux Instance
- Configure Security Groups
- Connect to instance with Putty
- Update OS, webserver, Query
- Security Group Update, Elastic IP (pending)
- Resize instance

NOTE: Terminate instance and Delete Elastic IP address (to avoid charge)

Demo – Windows Instance

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

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 instances access to other AWS services using IAM roles
- Launch Instance with that role
- Instance can automatically access the resources based on privileges granted in the role
- No need to maintain Access Key / Secret Access Key pairs
- Note: you cannot assign a role to an existing instance.

Differences between VPC and EC2-Classic

Table: Differences between VPC and EC2-Classic

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

- Elastic IP Address – Static Public IPv4 Address that you can request AWS and attach to a network interface of an instance
- Elastic IP – Limited to 5 IPv4 addresses per region per account
- Unused Elastic IP Addresses (including stopped instances) are charged an hourly fee. Due to scarce nature of public IPv4 addresses. Release it when no longer needed

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)

Instance Purchasing Options

Instance Purchasing Option

- On-Demand Instances – Pay by the hour
- Reserved Instances – Capacity Reservation
  - 1 or 3 year term
  - Significant discount over on-demand pricing
- Scheduled Instances – Capacity Reservation
  - 1 year term
  - Instances available at specified recurring schedule
- Spot Instances – Opportunistic and Interruptible
  - Bid for unused instances at significant discount
  - Runs only if they are available and bid is above spot price

Instance Purchasing Options

- Dedicated Hosts – Meet Compliance requirements
  - Pay for a physical host – hourly or reservation
  - Fully dedicated for running your instances
  - Use your license Bring Your Own License (BYOL)
  - Note: EBS and other services are still on multi-tenant hardware
- Dedicated Instances
  - Instance run on a single tenant hardware
  - Pay by the hour
  - Other instances from your account may run on the same host

Reserved Instances

- Reserved at either region or availability zone (AZ) level
- When bought at AZ level, it guarantees capacity in that AZ
- Reserved instance discount is automatically applied to running instances that match the reservation attributes
- 1 or 3 year term – Cannot cancel. Pay for the full term
- Three payment options
    - No upfront
    - Partial upfront
    - All upfront

Reserved Instances

- Convertible
  - 3 year term
  - Exchange for different instance family, platform, scope and tenancy
  - Can be exchanged for same or higher payment option
- Standard
  - Single instance family, platform, scope and tenancy
  - 1 year or 3 year term
  - Change AZ within the same region, Scope of reservation from Region <-> AZ, EC2-VPC <-> EC2-Classic, Instance Size within the same instance type

Scheduled Instances

- Capacity reservation on a recurring schedule
- Daily, Weekly or Monthly basis with a specified start time and duration
- 1 year term – pay even if not used. Minimum 1,200 hours per year
- Reserve from available EC2 pool dedicated for scheduled instances
- Terminates 3 minutes before the end of current schedule time period.

Spot Instances

- Bid on unused EC2 instances - Hourly price is set by EC2 (instance type/size)
- Fluctuates based on supply and demand
- Instance runs when bid exceeds current market price
- Can also bid for a spot instance fleet
- Terminated (2 minute notice) – market price exceeds bid price or capacity constraints or fleet constraints
- Well suited for data analysis, batch jobs, background processing and optional tasks – that can be restarted or continued from where it was left off

Dedicated Hosts

- Physical server with EC2 instance capacity fully dedicated to your use
- Allows you use your own license for Windows Server, SQL Server, SUSE, Linux Enterprise Server and so forth
- Pay hourly – on demand rate or reserve
- Rate depends on instance types that dedicated host supports and region
- No extra charge for running your instances on the host
- To terminate – stop all instances and then release the host

Dedicated Host Reservation

- Reserve dedicated host for 1 year or 3 year terms
  - No Upfront – 1 year term only
  - Partial Upfront – 1 or 3 year terms
  - Full Upfront – 1 or 3 year terms
- Cannot cancel – need to pay for the full term

Dedicated Host Restrictions

- Only BYOL Red Hat, SUSE, Windows AMIs offered by AWS or AWS marketplace can be used
- EC2 Auto recovery not supported
- Limit - Up to two on-demand dedicated host per instance family, region. Possible to request for limit increase
- Placement groups, Auto scaling groups, Managed RDS instances are not supported
- No. of instances you can launch depends on host

Dedicated Instances

- Launch instances on a hardware dedicated to your account
- Other non-dedicated instances that you launch with the same account may run on the same hardware
- Similar to dedicated host in host isolation; with dedicated host you can choose instance placement
- You can change instance tenancy from dedicated <-> host. But instance with default tenancy (shared hardware) cannot be changed

EC2 Pricing

- AWS Pricing Calculator
- Instance Pricing (on-demand vs reserved), Dedicated Instances
- Dedicated Hosts Pricing
- Data Transfer IN
- Data Transfer OUT (same region, different region, internet)
- EBS Optimization add-on pricing (some instance types are already enabled at no extra cost)

EC2 Data Transfer Pricing

- Data Transfer in a single AZ
  - Private IPv4 – free
  - Public or Elastic IP – charged at intra-regional data transfer rates
- Data Transfer across AZ in a single Region
  - Private or Public IP - charged at intra-regional data transfer rates
- Peered VPC – charged at intra-regional data transfer rates

EC2 Data Transfer Pricing

- Across regions – charged at inter-region transfer rates
- To Internet – charged at internet transfer rates
- From Internet - free
- Applies to EC2, RDS, Redshift, ElastiCache, ELB

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