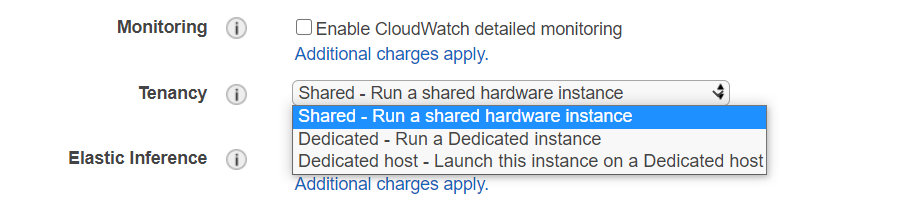
AWS EC2 Instance Purchasing Option

AWS EC2 Instance Purchasing Option

* Amazon provides different ways to pay for the EC2 instances
  + **On-Demand Instances**
  + **Reserved Instances**
  + **Spot Instances**
  + **Dedicated Hosts**
  + **Dedicated Instances**
* AWS also allows launching EC2 instance on **shared or dedicated tenancy**



* Light, Medium, and Heavy Utilization Reserved Instances are no longer available for purchase and were part of the Previous Generation AWS EC2 purchasing model

A screenshot of a computer

Description automatically generated with medium confidence

On-Demand Instances

* Pay for the instances and the compute capacity used` by the hour or the second, depending on which instances you run,
* No long-term commitments or up-front payments
* Instances can be scaled accordingly as per the demand
* Although AWS makes effort to have the capacity to launch On-Demand instances, there might be instances during peak demand where the instance cannot be launched
* Well suited for
  + Users that want the low cost and flexibility of EC2 without any up-front payment or long-term commitment
  + Applications with **short term, spiky, or unpredictable workloads that cannot be interrupted**
  + Applications being developed or tested on EC2 for the first time

Reserved Instances

* Reserved Instances provides lower hourly running costs by providing a billing discount (up to 75%) as well as capacity reservation that is applied to instances and there would **never be a case of insufficient capacity**
* Discounted usage price is fixed as long as you own the Reserved Instance, allowing compute costs prediction over the term of the reservation
* Reserved instances are best suited if **consistent, heavy, use is expected,** and they can provide savings over owning the hardware or running only On-Demand instances.
* Well Suited for
  + Applications with **steady state or predictable usage**
  + Applications that **require reserved capacity**
  + Users able to make upfront payments to reduce their **total computing costs even further**
* Reserved instance is **not a physical instance** that is launched, but rather a **billing discount** applied to the use of On-Demand Instances
* On-Demand Instances **must match** certain attributes, such as **instance type and Region**, in order to benefit from the billing discount.
* Reserved Instances do not renew automatically, and the EC2 instances can be continued to be used but charged On-Demand rates
* Auto Scaling or other AWS services can be used to launch the On-Demand instances that use the Reserved Instance benefits
* With Reserved Instances
  + **You pay for the entire term, regardless of the usage**
  + Once purchased, the **reservation cannot be cancelled** but can be sold in the Reserved Instance Marketplace
  + Reserved Instance pricing tier discounts only apply to purchases made from AWS, and not to the third party Reserved instances

Reserved Instance pricing key variables

Instance attributes

A Reserved Instance has **four instance attributes** that determine its price.

* **Instance type**: Instance family + Instance size e.g. m4.large composed of the instance family (m4) and the instance size (large).
* **Region**: Region in which the Reserved Instance is purchased.
* **Tenancy**: Whether your instance runs on shared (default) or single-tenant (dedicated) hardware.
* **Platform**: Operating system; for example, Windows or Linux/Unix.

Term commitment

Reserved Instance can be purchased for a **one-year or three-year commitment**, with the three-year commitment offering a bigger discount.

* **One-year**: A year is defined as 31536000 seconds (365 days).
* **Three-year**: Three years is defined as 94608000 seconds (1095 days).

Payment options

* **No Upfront**
  + No upfront payment is required, and the account is charged on a discounted hourly rate for every hour, regardless of the usage
  + Only available as 1-year reservation
* **Partial Upfront**
  + A portion of the cost is paid upfront and the remaining hours in the term are charged at an hourly discounted rate, regardless of the usage
* **Full Upfront**
  + Full payment is made at the start of the term, with no costs for the remainder of the term, regardless of the usage

Offering class

* **Standard**: Provide the most significant discount but can only be modified.
* **Convertible**: These provide a lower discount than Standard Reserved Instances but can be exchanged for another Convertible Reserved Instance with different instance attributes. **Convertible Reserved Instances can also be modified**.

Diagram

Description automatically generated

How Reserved Instances work

Billing Benefits & Payment Options

* Reserved Instance purchase reservation is automatically applied to running instances that match the specified parameters

Diagram

Description automatically generated

* Reserved Instance can also be utilized by launching On-Demand instances with the same configuration as to the purchased reserved capacity

Understanding Hourly Billing

* Reserved Instances are billed for every hour during the term that you select, regardless of whether the instance is running or not.
* *Reservations and discounted rates only apply to one instance-hour per hour. If an instance restarts during the first hour of a reservation and runs for two hours before stopping, the first instance-hour is charged at the discounted rate but three instance-hours are charged at the On-Demand rate. If the instance restarts during one hour and again the next hour before running the remainder of the reservation, one instance-hour is charged at the On-Demand rate but the discounted rate is applied to previous and subsequent instance-hours.*

Consolidated Billing

* **Pricing benefits of Reserved Instances are shared when the purchasing account is part of a set of accounts billed under one consolidated billing payer account**
* **Consolidated billing account aggregates the list value of member accounts within a region.**
* When the list value of all active Reserved Instances for the consolidated billing account reaches a discount pricing tier, **any Reserved Instances purchased after this point by any member** of the consolidated billing account are charged at the discounted rate (as long as the list value for that consolidated account stays above the discount pricing tier threshold)

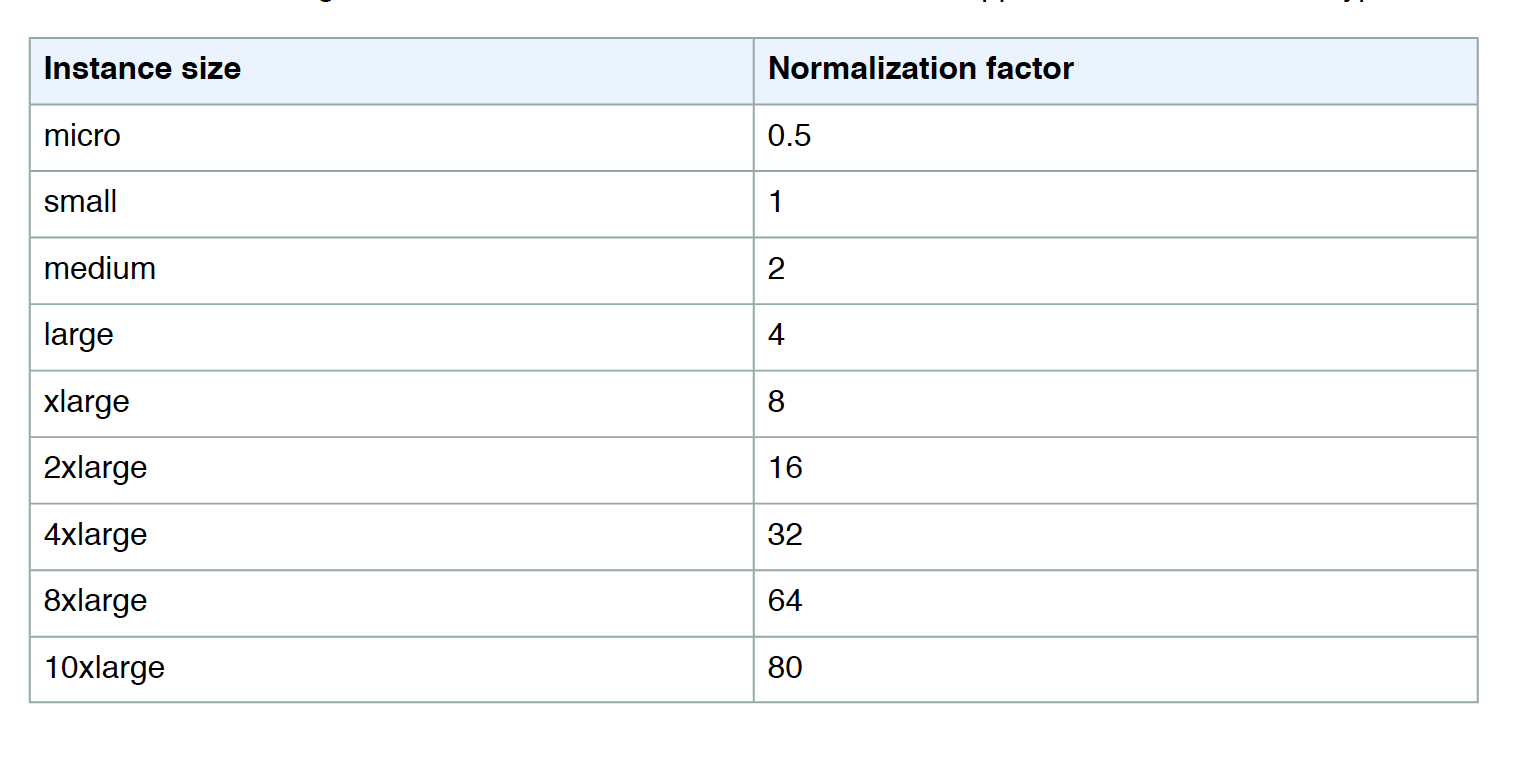
Buying Reserved Instances

Buying Reserved Instances need selection of the following

* **Platform** (for example, Linux)
* **Instance type** (for example, m1.small)
* **Availability Zone** in which to run the instance
* **Term** (time period) over which you want to reserve capacity
* **Tenancy** You can reserve capacity for your instance to run in single-tenant hardware (dedicated tenancy, as opposed to shared).
* **Offering** (No Upfront, Partial Upfront, All Upfront).

Modifying Reserved Instances

* Standard or Convertible Reserved Instances **can be modified** and continue to benefit from the capacity reservation as the computing needs change.
* **Availability Zone, instance size (within the same instance family), and scope** of the Reserved Instance can be modified
* **All or a subset of the Reserved Instances** can be modified
* Two or more Reserved Instances can be merged into a single Reserved Instance
* Modification does not change the remaining term of the Reserved Instances; their end dates remain the same.
* There is no fee, and you do not receive any new bills or invoices.
* Modification is separate from purchasing and does not affect how you use, purchase, or sell Reserved Instances.
* Complete reservation or a subset of it can be modified in one or more of the following ways:
  + Switch Availability Zones within the same region
  + Change between EC2-VPC and EC2-Classic
  + Change the instance size within the same instance type, given the instance size footprint remains the same *for e.g. four m1.medium instances (4 x 2), you can turn it into a reservation for eight m1.small instances (8 x 1) and vice versa. However, you cannot convert a reservation for a single m1.small instance (1 x 1) into a reservation for an m1.large instance (1 x 4).*



Scheduled Reserved Instances (Deprecating)

Graphical user interface, text, chat or text message

Description automatically generated

* Scheduled Reserved Instances (Scheduled Instances) enable capacity reservations purchase that recur on a daily, weekly, or monthly basis, with a specified start time and duration, for a one-year term.
* Capacity is reserved in advance and is always available when needed
* Charges are incurred for the time that the instances are scheduled, even if they are not used
* Scheduled Instances are a good choice for workloads that **do not run continuously, but do run on a regular schedule***for e.g. weekly or monthly batch jobs*
* EC2 launches the instances, based on the launch specification during their scheduled time periods
* EC2 terminates the EC2 instances **three minutes before the end of the current scheduled time period** to ensure the capacity is available for any other Scheduled Instances it is reserved for.
* Scheduled Reserved instances **cannot be stopped of rebooted**, however they can be terminated and relaunched within minutes of termination
* Scheduled Reserved instances limits or restrictions
  + after purchase cannot be modified, canceled or resold
  + only supported instance types: C3, C4, M4, and R3
  + required term is 365 days (one year).
  + minimum required utilization is 1,200 hours per year
  + purchase up to three months in advance

Savings Plan

EC2 Savings plans => **Selected Region & Family**

A picture containing diagram

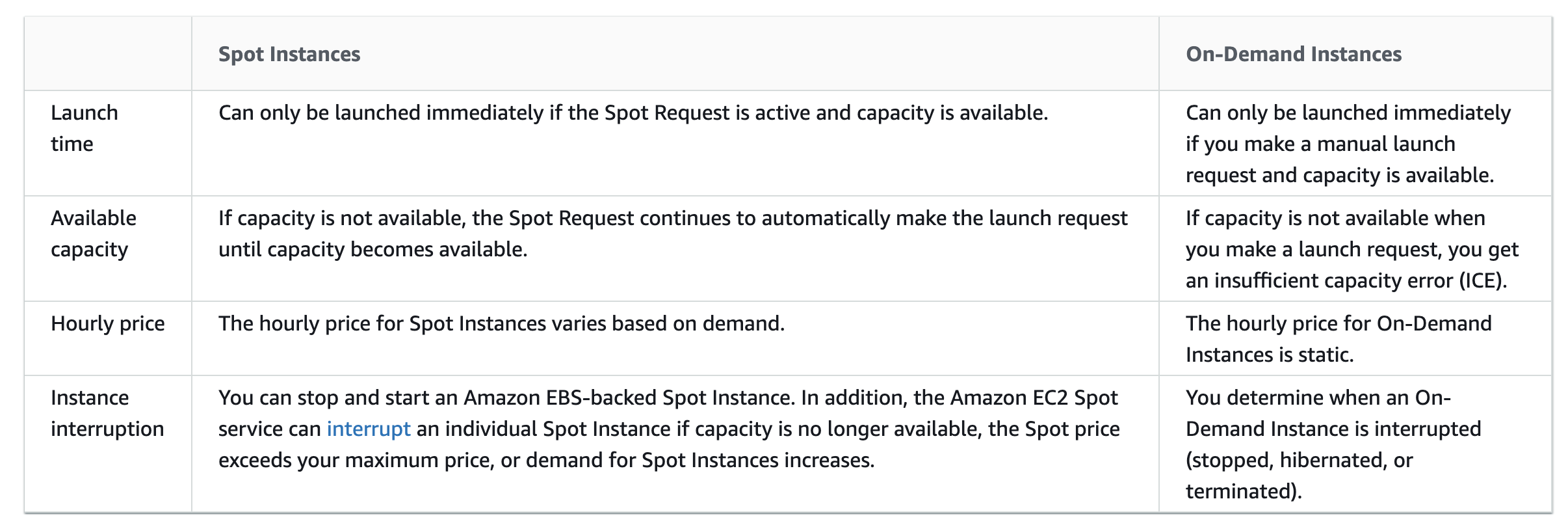
Description automatically generateds

Spot Instances

A picture containing text

Description automatically generated

* EC2 Spot instances allow access to spare EC2 computing capacity for up to 90% off the On-Demand price
* EC2 sets up the hourly price, referred to as **Spot price**, which fluctuates depending upon the demand and supply of spot instances
* Spot instances enables bidding on unused EC2 instances, and are launched whenever the bid price exceeds the current market spot price
* Spot instances are a cost-effective choice and can bring the EC2 costs down significantly
* Spot instances can be used for **applications flexible in the timing when they can run and also able to handle interruption** by storing the state externally *for e.g. they are well-suited for data analysis, batch jobs, background processing, and optional tasks*
* Spot instances differ from the On-Demand instances
  + they are not launched immediately
  + they can be terminated anytime
  + prices vary as per the demand and supply of spot instances
* **Usual strategy involves using Spot instances with On-Demand or Reserved instances, which provide a minimum level of guaranteed compute resources, while spot instances provide an additional computation boost**
* Spot instances can also be launched with a required duration (also known as **Spot blocks**), which are not interrupted due to changes in the Spot price
* EC2 provides a data feed, sent to an S3 bucket specified during subscription, that describes the Spot instance usage and pricing
* **T2 and HS1 instance class types are not supported for Spot instances**
* Well Suited for
  + Ideal for various stateless, fault-tolerant, or flexible applications such as big data, containerized workloads, CI/CD, high-performance computing (HPC), web servers, and other test & development workloads
  + Applications that have flexible start and end times
  + Applications that are only feasible at very low compute prices
  + Users with urgent computing needs for large amounts of additional capacity



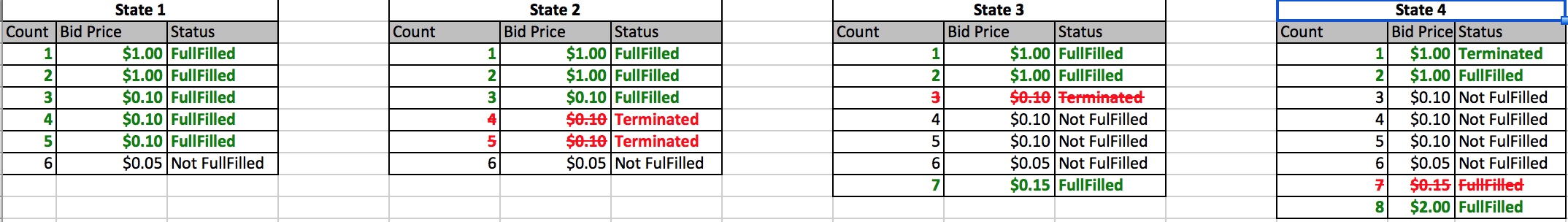
Spot Concepts

* **Spot pool** – Pool of EC2 instances with the same instance type, availability zone, operating system and network platform
* **Spot price** – Current market price of a spot instance per hour as set by EC2 based on the last fulfilled bid
* **Spot bid** – maximum bid price the bidder is willing to pay for the spot instance
* **Spot fleet** – set of instances launched based on the criteria the bidder
* **Spot Instance request** – Provides the maximum price per hour that you are willing to pay for a Spot Instance. If unspecified, it defaults to the On-Demand price. EC2 fulfils the request when the maximum price per hour for the request exceeds the Spot price and if capacity is available. A Spot Instance request is either **one-time or persistent**. EC2 automatically resubmits a persistent Spot request after the Spot Instance associated with the request is terminated. The Spot Instance request can optionally specify a duration for the Spot Instances.
* **Spot instance interruption** – EC2 terminates the spot instances whenever the bid price is lower than the current market price or the supply has reduced. EC2 provides a **Spot Instance interruption notice**, which gives the instance a **two-minute warning** before it is interrupted.
* **Bid status** – provides the current state of the spot bid

Spot Pricing & How it works

* EC2 sets up an hourly spot price which fluctuates depending upon the demand and supply.
* If the bid price exceeds the current market spot price, the request is fulfilled by Amazon till either the spot instance is terminated or the spot price increases beyond the bid price
* Everyone pays the same market price for the period irrespective of the bid price given the bid price is more than the spot price *for e.g. if the spot price is $0.20 and there are 2 bids from Customers with bid price $0.25 and $0.30, both customers would still pay $0.20 only*
* If the Spot instance is terminated by Amazon, you are not billed for the partial hour. However, if the spot instance are terminated by you, you will be charged for the partial hour
* Spot instances with a predefined duration use a fixed hourly price that remains in effect for the Spot instance while it runs
* EC2 can interrupt the Spot instance when the Spot price rises above your bid price, when the demand for Spot instances rises, or when the supply of Spot instances decreases.
* When EC2 marks a Spot instance for termination, it provides a Spot instance termination notice, which gives the instance a **two-minute warning** before it terminates.
* Termination notice warning is made available to the applications on the Spot instance using an item in the **instance metadata termination-time attribute** **http://169.254.169.254/latest/meta-data/spot/termination-time** and includes the time when the shutdown signal will be sent to the instance’s operating system
* Relevant applications on Spot Instances should poll for the termination notice at 5 second intervals, giving it almost the entire two minutes to complete any needed processing before the instance is terminated and taken back by AWS
* EBS-backed Spot Instance can be stopped – started, rebooted or terminated

**Pricing Example**



* **State 1** – Starting with Amazon EC2 has 5 Spot instances available
  + 6 bids available for Spot instances
  + Amazon EC2 picks up the top five priced bids and allocates a Spot instance to them
  + Spot Price is $0.10
  + Bid with the price of $0.05 is not served
* **State 2** – Supply of Amazon EC2 Spot instances reduce to 3
  + Amazon EC2 terminates the 2 spot instances with $0.10 ( the order in which the instances are terminated is determined at random )
  + Rest of the Spot instances continue
* **State 3** – New bid for Spot Instance is placed with Price $0.15 is placed
  + Spot Price changed to $0.15
  + Amazon EC2 terminates the single spot instances with $0.10
  + Spot instance with price $0.15 is fulfilled
* **State 4** New bid for Spot Instance is placed with Price $2 is placed
  + Spot instance with price $2 is fulfilled
  + Amazon EC2 terminates the single spot instances with $0.15
  + Spot Price changed to $1.00

Spot Fleets

* Spot Fleet is a collection, or fleet, of Spot Instances, and optionally On-Demand Instances
* Spot Fleet attempts to launch the number of Spot Instances and On-Demand Instances to meet the specified target capacity
* Request for Spot Instances is fulfilled if there is available capacity and the maximum price specified in the request exceeds the current Spot price
* Spot Fleet also attempts to maintain its target capacity fleet if the Spot Instances are interrupted.

Spot Block

Timeline

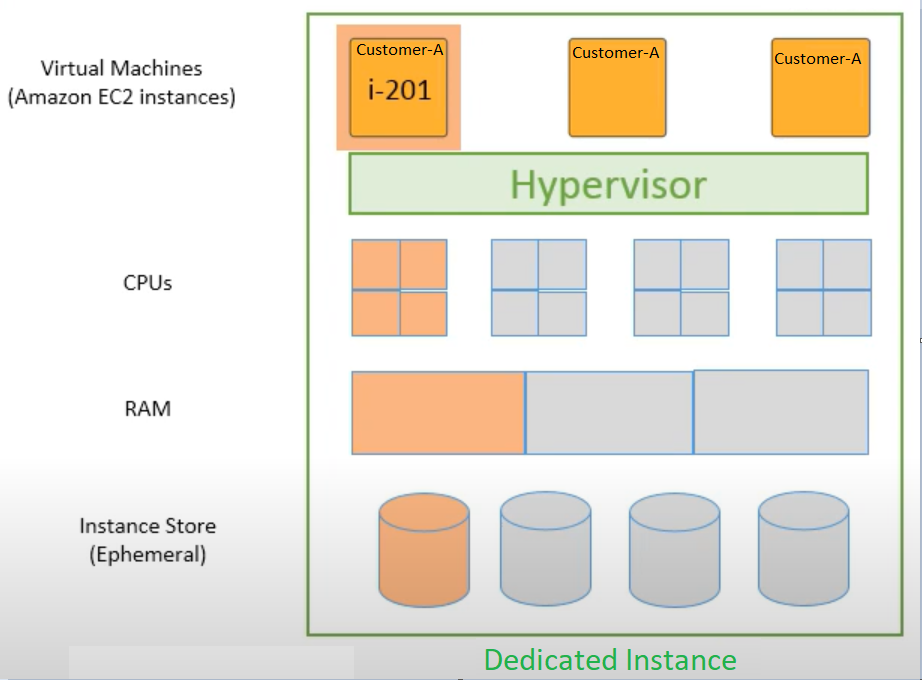
Description automatically generated

Spot Instances best practices

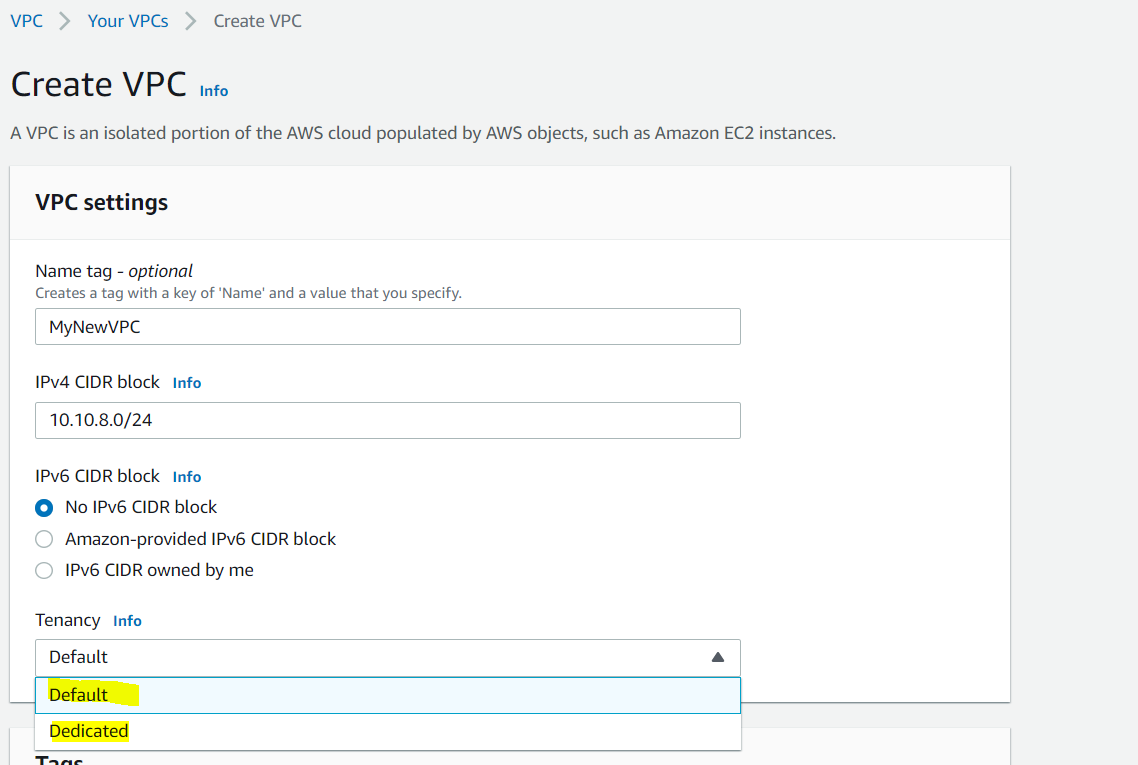
* **Choose a reasonable bid price**, which is low enough to suit the budget and high enough for the request to be fulfilled and should not be higher than the On-Demand bid price
* **Be flexible about instance types and Availability Zones**–  A Spot Instance pool is a set of unused EC2 instances with the same instance type (for example, m5.large) and AZ (for example, us-east-1a). Be flexible about which instance types you request and in which AZs you can deploy your workload. This gives Spot a better chance to find and allocate your required amount of compute capacity.
* **Ensure the instances are up and ready as soon as the request is fulfilled**, by provisioning an AMI with all the required softwares and load application data from user data
* **Prepare individual instances for interruptions –**Make application to be fault-tolerant. Store important data regularly and externally in a place that won’t be affected by Spot instance termination *for e.g., use S3, EBS, or DynamoDB.*
* **Divide the work into smaller finer tasks** so that they can be completely and state saved more frequently
* **Use Spot termination notice warning** to monitor instance status regularly
* **Test applications** using On-Demand instances and terminating them to ensure that it handles unexpected termination gracefully

EC2 Dedicated Instances

* Dedicated Instances are EC2 instances that run in a VPC on hardware that’s dedicated to a single customer
* Dedicated Instances are **physically isolated at the host hardware level** from the instances that aren’t Dedicated Instances and from instances that belong to other AWS accounts.

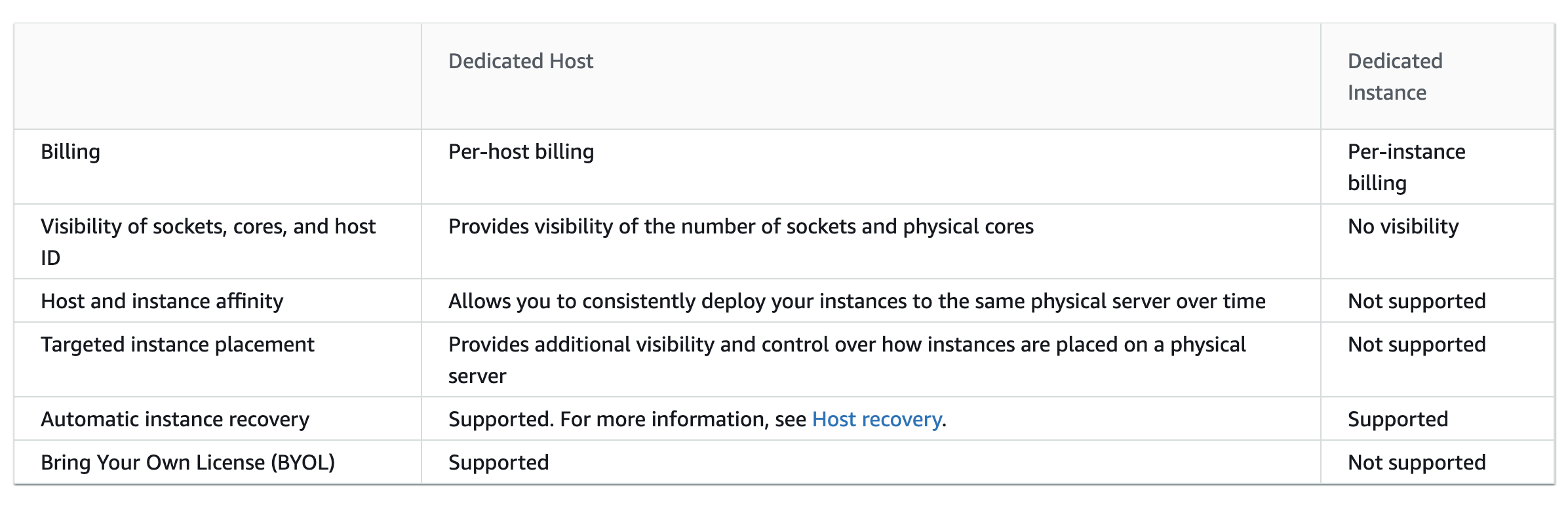


* Affinity – Stop/Start instance might bring instance on a different dedicated instance.
* Each VPC has a related instance tenancy attribute, and can’t be changed after the VPC has been created. Default is shared.
* Each instance launched into a VPC has a tenancy attribute, can’t be changed after the instance is launched. Default tenancy depends on the VPC tenancy, which by default is shared.
* Dedicated Instances can be launched using
  + Create the VPC with the instance tenancy set to dedicated, all instances launched into this VPC are Dedicated Instances even though if you mark the tenancy as shared
  + Create the VPC with the instance tenancy set to default, and specify dedicated tenancy for any instances that should be Dedicated Instances when you launch them.



Dedicated Hosts

* EC2 Dedicated Host is a **physical server with EC2 instance capacity fully dedicated** to your use
* **Dedicated Hosts allow using existing per-socket, per-core, or per-VM software licenses, including Windows Server, Microsoft SQL Server, SUSE, and Linux Enterprise Server.**
* **Supports Bring your own licenses (BYOL).**
* **Affinity – Stop/Start instance will keep instance on same dedicated host.**



A picture containing text

Description automatically generated

Timeline

Description automatically generated

Timeline

Description automatically generated

AWS Certification Exam Practice Questions

* Questions are collected from Internet and the answers are marked as per my knowledge and understanding (which might differ with yours).
* AWS services are updated everyday and both the answers and questions might be outdated soon, so research accordingly.
* AWS exam questions are not updated to keep up the pace with AWS updates, so even if the underlying feature has changed the question might not be updated
* Open to further feedback, discussion and correction.

1. If I want my instance to run on a single-tenant hardware, which value do I have to set the instance’s tenancy attribute to?
   1. **dedicated**
   2. isolated
   3. one
   4. reserved
2. You have a video transcoding application running on Amazon EC2. Each instance polls a queue to find out which video should be transcoded, and then runs a transcoding process. If this process is interrupted, the video will be transcoded by another instance based on the queuing system. You have a large backlog of videos, which need to be transcoded, and would like to reduce this backlog by adding more instances. You will need these instances only until the backlog is reduced. Which type of Amazon EC2 instances should you use to reduce the backlog in the most cost efficient way?
   1. Reserved instances
   2. **Spot instances**
   3. Dedicated instances
   4. On-demand instances
3. The one-time payment for Reserved Instances is \_\_\_\_\_\_\_\_\_\_ refundable if the reservation is cancelled.
   1. always
   2. in some circumstances
   3. **never**
4. You run a web application where web servers on EC2 Instances are In an Auto Scaling group Monitoring over the last 6 months shows that 6 web servers are necessary to handle the minimum load. During the day up to 12 servers are needed Five to six days per year, the number of web servers required might go up to 15. What would you recommend to minimize costs while being able to provide hill availability?
   1. **6 Reserved instances (heavy utilization). 6 Reserved instances (medium utilization), rest covered by On-Demand instances**
   2. 6 Reserved instances (heavy utilization). 6 On-Demand instances, rest covered by Spot Instances (don’t go for spot as availability not guaranteed)
   3. 6 Reserved instances (heavy utilization) 6 Spot instances, rest covered by On-Demand instances (don’t go for spot as availability not guaranteed)
   4. 6 Reserved instances (heavy utilization) 6 Reserved instances (medium utilization) rest covered by Spot instances (don’t go for spot as availability not guaranteed)
5. A user is running one instance for only 3 hours every day. The user wants to save some cost with the instance. Which of the below mentioned Reserved Instance categories is advised in this case?
   1. **The user should not use RI; instead only go with the on-demand pricing**(seems question before the introduction of the Scheduled Reserved instances in Jan 2016, which can be used in this case)
   2. The user should use the AWS high utilized RI
   3. The user should use the AWS medium utilized RI
   4. The user should use the AWS low utilized RI
6. Which of the following are characteristics of a reserved instance? Choose 3 answers (but 4 answers seem correct)
   1. **It can be migrated across Availability Zones** (can be modified)
   2. It is specific to an Amazon Machine Image (AMI) (specific to platform)
   3. **It can be applied to instances launched by Auto Scaling**(are allowed)
   4. It is specific to an instance Type (specific to instance family but instance type can be changed)
   5. **It can be used to lower Total Cost of Ownership (TCO) of a system**(helps to reduce cost)
7. You have a distributed application that periodically processes large volumes of data across multiple Amazon EC2 Instances. The application is designed to recover gracefully from Amazon EC2 instance failures. You are required to accomplish this task in the most cost-effective way. Which of the following will meet your requirements?
   1. **Spot Instances**
   2. Reserved instances
   3. Dedicated instances
   4. On-Demand instances
8. Can I move a Reserved Instance from one Region to another?
   1. **No**
   2. Only if they are moving into GovCloud
   3. Yes
   4. Only if they are moving to US East from another region
9. An application you maintain consists of multiple EC2 instances in a default tenancy VPC. This application has undergone an internal audit and has been determined to require dedicated hardware for one instance. Your compliance team has given you a week to move this instance to single-tenant hardware. Which process will have minimal impact on your application while complying with this requirement?
   1. Create a new VPC with tenancy=dedicated and migrate to the new VPC (possible but impact not minimal)
   2. Use ec2-reboot-instances command line and set the parameter “dedicated=true”
   3. Right click on the instance, select properties and check the box for dedicated tenancy
   4. **Stop the instance, create an AMI, launch a new instance with tenancy=dedicated, and terminate the old instance**
10. Your department creates regular analytics reports from your company’s log files. All log data is collected in Amazon S3 and processed by daily Amazon Elastic Map Reduce (EMR) jobs that generate daily PDF reports and aggregated tables in CSV format for an Amazon Redshift data warehouse. Your CFO requests that you optimize the cost structure for this system. Which of the following alternatives will lower costs without compromising average performance of the system or data integrity for the raw data? **[PROFESSIONAL]**
    1. Use reduced redundancy storage (RRS) for PDF and CSV data in Amazon S3. Add Spot instances to Amazon EMR jobs. Use Reserved Instances for Amazon Redshift. (Spot instances impacts performance)
    2. **Use reduced redundancy storage (RRS) for all data in S3. Use a combination of Spot instances and Reserved Instances for Amazon EMR jobs. Use Reserved instances for Amazon Redshift**(Combination of the Spot and reserved with guarantee performance and help reduce cost. Also, RRS would reduce cost and guarantee data integrity, which is different from data durability )
    3. Use reduced redundancy storage (RRS) for all data in Amazon S3. Add Spot Instances to Amazon EMR jobs. Use Reserved Instances for Amazon Redshift (Spot instances impacts performance)
    4. Use reduced redundancy storage (RRS) for PDF and CSV data in S3. Add Spot Instances to EMR jobs. Use Spot Instances for Amazon Redshift. (Spot instances impacts performance)
11. A research scientist is planning for the one-time launch of an Elastic MapReduce cluster and is encouraged by her manager to minimize the costs. The cluster is designed to ingest 200TB of genomics data with a total of 100 Amazon EC2 instances and is expected to run for around four hours. The resulting data set must be stored temporarily until archived into an Amazon RDS Oracle instance. Which option will help save the most money while meeting requirements? **[PROFESSIONAL]**
    1. **Store ingest and output files in Amazon S3. Deploy on-demand for the master and core nodes and spot for the task nodes.**
    2. Optimize by deploying a combination of on-demand, RI and spot-pricing models for the master, core and task nodes. Store ingest and output files in Amazon S3 with a lifecycle policy that archives them to Amazon Glacier. (Reserved Instance not cost effective for 4 hour job and data not needed in S3 once moved to RDS)
    3. Store the ingest files in Amazon S3 RRS and store the output files in S3. Deploy Reserved Instances for the master and core nodes and on-demand for the task nodes. (Reserved Instance not cost effective)
    4. Deploy on-demand master, core and task nodes and store ingest and output files in Amazon S3 RRS (RRS provides not much cost benefits for a 4 hour job while the amount of input data would take time to upload and Output data to reproduce)
12. A company currently has a highly available web application running in production. The application’s web front-end utilizes an Elastic Load Balancer and Auto scaling across 3 availability zones. During peak load, your web servers operate at 90% utilization and leverage a combination of heavy utilization reserved instances for steady state load and on-demand and spot instances for peak load. You are asked with designing a cost effective architecture to allow the application to recover quickly in the event that an availability zone is unavailable during peak load. Which option provides the most cost effective high availability architectural design for this application? **[PROFESSIONAL]**
    1. **Increase auto scaling capacity and scaling thresholds to allow the web-front to cost-effectively scale across all availability zones to lower aggregate utilization levels that will allow an availability zone to fail during peak load without affecting the applications availability.**(Ideal for HA to reduce and distribute load)
    2. Continue to run your web front-end at 90% utilization, but purchase an appropriate number of utilization RIs in each availability zone to cover the loss of any of the other availability zones during peak load. (90% is not recommended as well RIs would increase the cost)
    3. Continue to run your web front-end at 90% utilization, but leverage a high bid price strategy to cover the loss of any of the other availability zones during peak load. (90% is not recommended as high bid price would not guarantee instances and would increase cost)
    4. Increase use of spot instances to cost effectively to scale the web front-end across all availability zones to lower aggregate utilization levels that will allow an availability zone to fail during peak load without affecting the applications availability. (Availability cannot be guaranteed)
13. You run accounting software in the AWS cloud. This software needs to be online continuously during the day every day of the week, and has a very static requirement for compute resources. You also have other, unrelated batch jobs that need to run once per day at any time of your choosing. How should you minimize cost? **[PROFESSIONAL]**
    1. **Purchase a Heavy Utilization Reserved Instance to run the accounting software. Turn it off after hours. Run the batch jobs with the same instance class, so the Reserved Instance credits are also applied to the batch jobs.**(Because the instance will always be online during the day, in a predictable manner, and there are sequences of batch jobs to perform at any time, we should run the batch jobs when the account software is off. We can achieve Heavy Utilization by alternating these times, so we should purchase the reservation as such, as this represents the lowest cost. There is no such thing a “Full” level utilization purchases on EC2.)
    2. Purchase a Medium Utilization Reserved Instance to run the accounting software. Turn it off after hours. Run the batch jobs with the same instance class, so the Reserved Instance credits are also applied to the batch jobs.
    3. Purchase a Light Utilization Reserved Instance to run the accounting software. Turn it off after hours. Run the batch jobs with the same instance class, so the Reserved Instance credits are also applied to the batch jobs.
    4. Purchase a Full Utilization Reserved Instance to run the accounting software. Turn it off after hours. Run the batch jobs with the same instance class, so the Reserved Instance credits are also applied to the batch jobs.