Cloud Instances and Auto-Scaling - Part 2

This lesson discusses cloud instance types and their life cycles.

We'll cover the following

Instances are classified based on their hardware capabilities, such as the CPU power, disk storage capacity, memory size, and so on.

Below are the instance types that are commonly offered by cloud providers.

Types of instances

Standard instances

These instances are configured for general use cases, including running web apps, microservices, etc. They have a balance between the resources allocated to them such as CPU, memory, and so on. Standard instances fit best for hosting general compute workloads.

High-CPU instances

These instances are specifically built to provide the high computing power required to run compute-intensive workloads such as distributed analytics, batch processing, machine learning algorithms, scalable multi-player gaming, graphics rendering, etc.

High-memory instances

These instances are built for running memory-intensive workloads such as real-time data ingestion, big data analytics, high-performance databases, running distributed in-memory caches & so on.

Instances with GPU

These instances provide the power of GPU for advanced computing requirements such as running data-intensive machine learning algorithms, data processing, 3D rendering, animation, virtual reality applications, autonomous vehicles, fluid dynamics, blockchain computations, and so on.

Pre-emptible instances

Pre-emptible instances are instances that are offered at a lower rate than the rate of regular instances by the cloud provider.

Lower rate? Why?

Their availability is not guaranteed at all times, and cloud providers offer these instances based on their availability. They can be terminated at any time from our workload and can be allocated to other high priority tasks.

Wait, what?

The provider can pull out instances running my service at any time? Why on earth would I ever opt for such kinds of instances?

Well, often we have use cases that are not really mission-critical and high priority like image processing, analytics, file processing, etc. These processes are batch processes, and they run in the background like a daemon.

Pre-emptible instances can be used for running these tasks, even if the instances are terminated and pulled away. The slowing down of these tasks does not impact

the service much as a whole, and the business can save money in the process.

Instances and storage

Ephemeral storage

Instances also have memory attached to them, known as the *ephemeral storage*, which can be augmented based on the requirement. This helps retain the running state in case the instance goes down due to failure or reboots. This memory is local to the compute instance and is purged after the instance is terminated.

Ephemeral storage is ideal for storing temporary data, including such as the local cache, buffers, OS, runtime data, and so on.

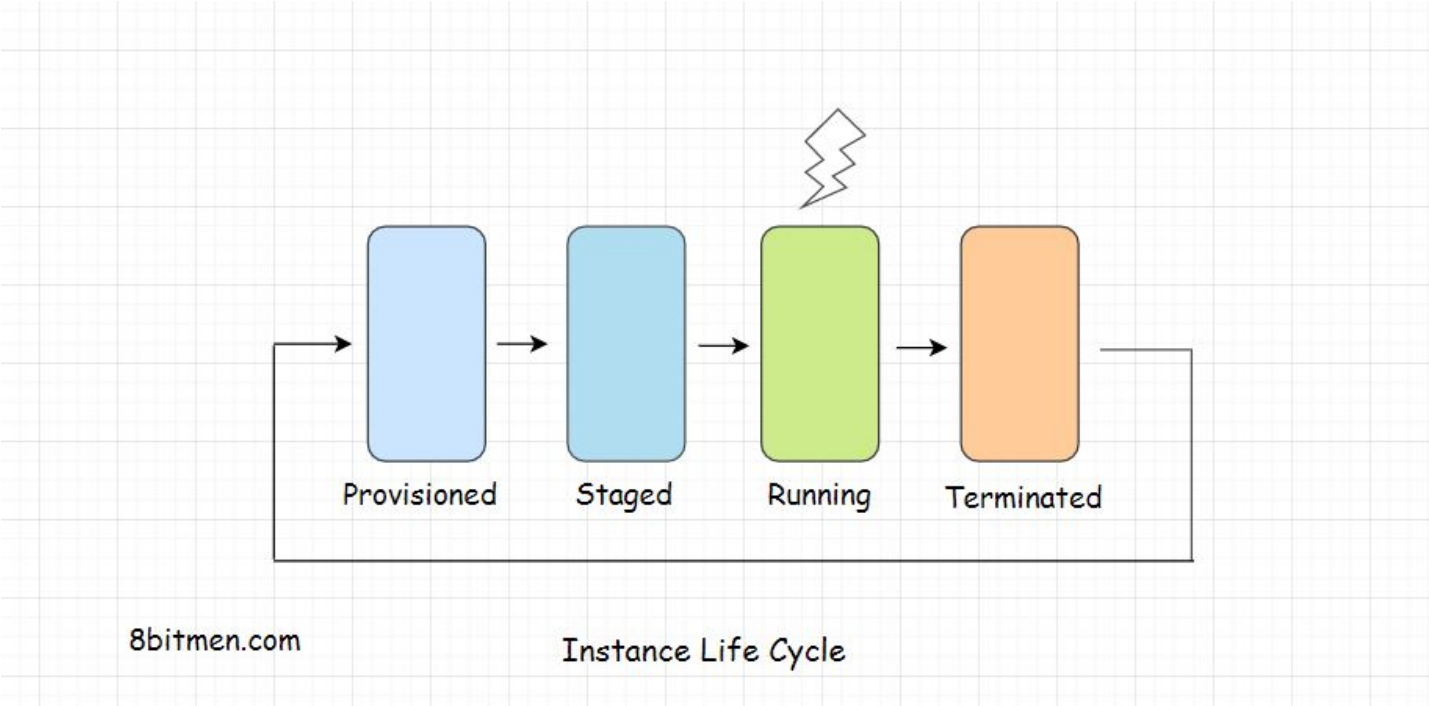
Persistent storage

There is another type of storage known as *persistent storage*. The scope of this memory is beyond individual compute instances. Newly spun up instances in the cluster can easily access the running state of the cluster from the persistent storage and continue the task without having the end-user notice the instance swap.

Instance life cycle

Right from the point an instance is provisioned, it goes through various stages in its life cycle such as being provisioned, staged, in the running state, terminated, and spun up again.

Let's quickly go through the different stages in an instance's life-cycle.



Provisioned

In this stage, the instances are not running yet. They are allocated to run a workload based on the configuration.

Staged

Cloud resources get allocated to an instance in this stage. The instance gets prepped up for the launch.

Running

The instance starts hosting the workload. If multiple instances are already running in the cluster, the new instance shares the load with the other already running instances.

Terminated

The instance is down either due to failure or manually done by the user. It can be reset, restarted, or deleted at this stage.

Let’s continue this discussion on instances in the next lesson.