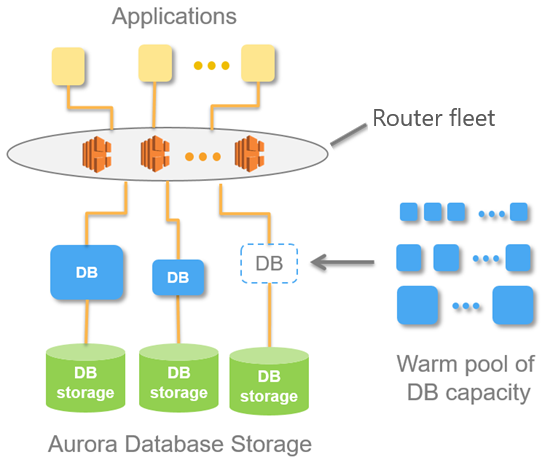
AWS RDS Aurora Serverless

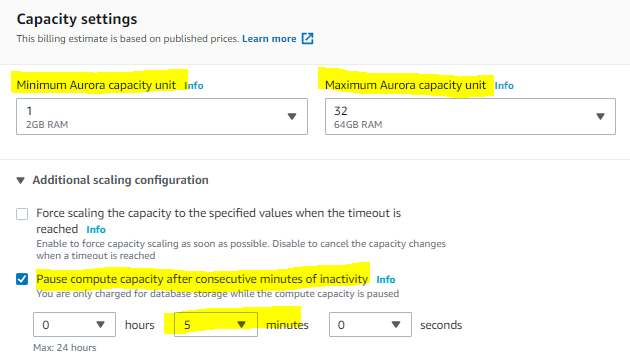
## Aurora Serverless

* Amazon Aurora Serverless is an on-demand, autoscaling configuration for the MySQL-compatible and PostgreSQL-compatible editions of Aurora.
* An Aurora Serverless DB cluster automatically starts up, shuts down, and scales capacity up or down based on the application’s needs.
* Aurora Serverless enables running database in the cloud without managing any database instances.
* Aurora Serverless provides a relatively simple, cost-effective option for infrequent, intermittent, or unpredictable workloads.
* Aurora Serverless Use Cases include
  + Infrequently-Used Applications
  + New Applications – where the needs and instance size is yet to be determined.
  + Variable and Unpredictable Workloads – scale as per the needs
  + Development and Test Databases
  + Multitenant Applications
* **Aurora Serverless DB cluster does not have a public IP address and can be accessed only from within a VPC based on the VPC service.**

## Aurora Architecture



* Aurora Serverless separates Storage and Compute, so it can scale down to zero processing and you pay only for storage.
* With Aurora Serverless, a database endpoint is created without specifying the DB instance class size.
* Minimum and maximum capacity is set in terms of Aurora capacity units (ACUs). Each ACU is a combination of processing and memory capacity.
* Database storage automatically scales from 10 GiB to 64 TiB, the same as storage in a standard Aurora DB cluster.
* The minimum Aurora capacity unit is the lowest ACU to which the DB cluster can scale down. The maximum Aurora capacity unit is the highest ACU to which the DB cluster can scale up. Based on the settings, Aurora Serverless automatically creates scaling rules for thresholds for CPU utilization, connections, and available memory.
* Database endpoint connects to a proxy fleet that routes the workload to a fleet of resources that are automatically scaled.
* Aurora Serverless manages the connections automatically.
* Proxy fleet enables continuous connections as Aurora Serverless scales the resources automatically based on the minimum and maximum capacity specifications.
* Database client applications don’t need to change to use the proxy fleet.
* Scaling is rapid because it uses a pool of “warm” resources that are always ready to service requests.
* Aurora Serverless supports **Automatic Pause** where the DB cluster can be paused after a given amount of time with no activity. The default inactvity timeout is five minutes. Pausing the DB cluster can be disabled.
* Automatic Pause reduces the compute charges to zero and only storage is charged. If database connections are requested when an Aurora Serverless DB cluster is paused, the DB cluster automatically resumes and services the connection requests.
* When the DB cluster resumes activity, it has the same capacity as it had when Aurora paused the cluster. The number of ACUs depends on how much Aurora scaled the cluster up or down before pausing it.



## Aurora Serverless and Failover

* Aurora Serverless compute layer is placed in a Single AZ
* Aurora separates computation capacity and storage, the storage volume for the cluster is spread across multiple AZs. The data remains available even if outages affect the DB instance or the associated AZ.
* Aurora Serverless supports **automatic multi-AZ failover** where if the DB instance for an DB cluster becomes unavailable or the Availability Zone (AZ) it is in fails, Aurora recreates the DB instance in a different AZ.
* Auroa Serverless failover mechanism takes longer than for an Aurora Provisioned cluster.
* Aurora Serverless failover time is currently undefined because it depends on demand and capacity availability in other AZs within the given AWS Region

## Aurora Serverless Auto Scaling

* Aurora Serverless automatically scales based on the active database workload ( CPU or connections), in some cases, capacity might not scale fast enough to meet a sudden workload change, such as a large number of new transactions.
* Once a scaling operation is initiated, Aurora Serverless attempts to find a scaling point, which is a point in time at which the database can safely complete scaling.
* Aurora Serverless might not be able to find a scaling point and will not scale if there are
  + long-running queries or transactions in progress, or
  + temporary tables or table locks in use.
* Supports cooldown period
  + After Scale up, Aurora Serverless has a 15 minutes cooldown period for subsequent scale down
  + After Scale down, Aurora Serverless has a 310 secs cooldown period for subsequent scale down
* Aurora Serverless has not cooldown period for scale up activities and scales as and when necessary.