**AWS database solutions for PostgreSQL comparison**

**AWS Aurora Serverless V1 (PostgreSQL)**

1. **Predictable Performance:**

* Serverless architecture automatically adjusts capacity based on actual usage, making it suitable for unpredictable workloads.
* Scales between a minimum and maximum capacity, providing flexibility.

1. **Steady Workloads:**

* Well-suited for variable workloads and applications with intermittent usage.
* May have warm-up times when it needs to scale up based on demand.

1. **Database Features/Capabilities:**

* Aurora features compatibility with PostgreSQL, high availability, and automatic backups.
* Limited control over certain database parameters compared to Aurora Provisioned.
* Must be provisioned with minimum 2 Aurora Capacity Units (ACU), doubles when scaling up.

**AWS Aurora Serverless V2 (PostgreSQL)**

1. **Predictable Performance:**

* Improved start-up times compared to V1, making it more responsive to fluctuations in demand.
* Automatic scaling based on workload, but with more granular capacity adjustments.

1. **Steady Workloads:**

* Suitable for variable workloads with more responsive scaling compared to V1.

1. **Database Features/Capabilities:**

* Inherits features from Aurora, including compatibility with PostgreSQL, high availability, and automatic backups.
* Improved control over database parameters compared to V1.
* Must be provisioned with minimum 0.5 Aurora Capacity Units (ACU), scales up every 0.5 ACU.

**AWS Aurora Provisioned (PostgreSQL)**

1. **Predictable Performance:**

* Fixed capacity based on the chosen instance type, providing more predictable performance for steady workloads.
* Can handle high-performance requirements.

1. **Steady Workloads:**

* Best suited for applications with consistent and high-performance requirements.

1. **Database Features/Capabilities:**

* Full control over database parameters and configurations.
* Supports read replicas for scaling read operations.
* Must be provisioned with minimum “medium” instance type.

**AWS RDS Postgres**

1. **Predictable Performance:**

* Fixed capacity based on the chosen instance type.
* Offers different instance classes to meet varying performance requirements.

1. **Steady Workloads:**

* Suitable for applications with steady workloads and known performance requirements.

1. **Database Features/Capabilities:**

* Full control over PostgreSQL configurations.
* Supports Multi-AZ deployments for high availability.

**Cost Comparison:**

1. **Aurora Serverless V1/V2:**

* Costs are based on actual capacity usage, making it cost-effective for intermittent workloads.
* Serverless V1 – $0.07/ h / ACU, billed per second.
* Serverless V2 – $0.14 / h / ACU, billed per second.

1. **Aurora Provisioned:**

* Costs are fixed based on the provisioned capacity of the instance type, suitable for steady workloads.
* May have higher upfront costs but potentially lower operational costs for steady-state applications.
* The cheapest option in London region is db.t4g.medium - $0.082 / h.

1. **RDS Postgres:**

* Costs are based on the chosen instance type, offering flexibility.
* Can be cost-effective for steady workloads but may lack the auto-scaling features of Aurora Serverless.
* The cheapest option in London region is db.t4g.micro - $0.018.

**Ongoing Management, Scaling, Backup, and Maintenance:**

1. **Aurora Serverless V1/V2:**

* Automatic scaling, less manual intervention.
* Automated backups and high availability.

1. **Aurora Provisioned:**

* Manual scaling based on instance type.
* Automated backups, Multi-AZ deployments for high availability.

1. **RDS Postgres:**

* Manual scaling based on instance type.
* Automated backups, Multi-AZ deployments for high availability.

**Summary:**

1. **Use Aurora Serverless V1/V2 for:**

* Intermittent workloads with variable demand.
* Applications with unpredictable usage patterns.

1. **Use Aurora Provisioned for:**

* Steady workloads with consistent high-performance requirements.
* Applications with specific configuration needs.

1. **Use RDS Postgres for:**

* Steady workloads with known performance requirements.
* Applications requiring manual control over configurations.