**AWS Architecture for the Application**

**Core Components**

1. **Amazon RDS/Amazon DynamoDB**: For relational data, Amazon RDS can be used; for NoSQL, DynamoDB is suitable.
2. **Amazon S3**: For storing large datasets, backups, and logs.
3. **AWS Lambda**: To handle real-time data processing and batch processing tasks.
4. **Amazon Redshift/Snowflake**: For data warehousing solutions. Redshift if using AWS's native solution or Snowflake as an alternative.
5. **Amazon EC2**: To host application servers and other components that require a virtual server environment.
6. **Amazon API Gateway**: To create, publish, maintain, monitor, and secure APIs.

**Data Loading (Batch and Real-Time)**

* **Batch Loading**: Use AWS Glue or Lambda functions to periodically transfer data from RDS/DynamoDB to Redshift/Snowflake.
* **Real-Time Loading**: Implement Kinesis Streams for capturing and loading real-time data into the database. Lambda can be used for light processing and transferring data to the appropriate services.

**Resilience, Performance, and Security**

* **Auto-Scaling and Load Balancing**: Use EC2 Auto Scaling and Elastic Load Balancing to manage and distribute incoming application traffic, ensuring high performance.
* **Data Redundancy and Backup**: Implement S3 versioning and RDS snapshots for data backup and redundancy.
* **Security**: Utilize AWS Identity and Access Management (IAM) for secure access control. Employ encryption for data at rest (using RDS/DynamoDB encryption features) and in transit (using SSL/TLS).

**Advantages of Snowflake Over PostgreSQL in AWS**

**Snowflake Advantages**

1. **Separation of Compute and Storage**: Snowflake's architecture allows independent scaling of compute and storage, leading to cost-efficiency and performance optimization.
2. **Zero-Copy Cloning and Time Travel**: Enables easy data recovery and historical data analysis without additional storage costs.
3. **Data Sharing**: Facilitates secure and easy data sharing, beneficial for collaborative healthcare environments.
4. **Automatic Performance Tuning**: Reduces the need for manual performance tuning and maintenance.
5. **Native Support for Semi-Structured Data**: Efficiently handles JSON, Avro, XML directly, which is advantageous for varied healthcare data formats.

**Conclusion**

Transitioning to a NoSQL database architecture, like MongoDB or Neo4J, offers flexibility, scalability, and efficient handling of complex relationships, essential for a dynamic and data-intensive field like healthcare. Implementing the application in AWS provides a robust, scalable, and secure environment, leveraging services like Lambda, S3, and RDS/DynamoDB. Snowflake, as an alternative to PostgreSQL, brings additional benefits in terms of scalability, data recovery, and performance optimization, making it a compelling choice for modern healthcare data warehousing needs.