**Architecture and Process (including AWS Software and products) to house our application in AWS in a Resilient, Secure, and Performant manner.**

* **Resilience:**

**Multi-AZ Deployment:** For automated failover that guarantees high availability and resilience against instance failures, use Amazon RDS with a Multi-AZ setup.

**Disaster Recovery:** Utilize Cross-Region Replication (CRR) as part of a disaster recovery plan to establish database replicas in another AWS region.

**Automated Backups:** Set up daily automatic backups and store them for a reasonable amount of time. Before making any big changes, manually take snapshots so you can recover quickly.

**Amazon Route 53:** For DNS administration, use Amazon Route 53, which makes failover simple and directs traffic to operational database instances in the event of a problem.

* **High Performance:**

**Amazon Aurora:** With automated replication, interoperability with MySQL and PostgreSQL, and scalability, Amazon Aurora is a good option.

**Performance Insights:** To visualize database performance, pinpoint bottlenecks, and enhance query performance, enable RDS Performance Insights.

**Amazon CloudWatch:** To guarantee the general health and functionality of the system, use CloudWatch for real-time monitoring and alert setup for important metrics.

* **Security:**

**Encryption:** Use RDS to enable encryption while sensitive medical data is in transit and at rest to prevent unwanted access.

**VPC Isolation:** Set up private subnets, restrict access with security groups and network ACLs, and use Amazon VPC to separate the database from the public internet.

**IAM Roles:** To securely manage access to AWS resources, implement IAM roles and rules while adhering to the least privilege principle.

**Secrets Manager:** Centralize and safeguard critical data by securely storing and managing database credentials using AWS Secrets Manager.

**AWS WAF and Shield:** Use AWS online Application Firewall (WAF) and Shield to defend against Distributed Denial of Service (DDoS) attacks and popular online vulnerabilities.

**AWS Key Management Service (KMS):** This service offers fine-grained control over key rules and is useful for securely managing encryption keys.

**Continuous Compliance:** To guarantee ongoing adherence to industry rules and security best practices, use AWS Config and Config Rules.

**Incident Response:** Establish incident response strategies to promptly address security incidents. Prepare for incident response by utilizing AWS Incident Manager.

**Secure Access:** To gain secure access to on-premises infrastructure, create a dedicated network connection or VPN tunnel using AWS Direct Connect or VPN.

**Well-Architected Framework:** Utilize the AWS Well-Architected Framework to regularly assess the architecture and make sure it complies with best practices for security, dependability, performance efficiency, cost optimization, and operational excellence.

These components are included in the design to provide excellent performance for database operations, robustness against faults, and strong security measures to safeguard private medical data.

