**Reasons for Implementation Choices:**

* Chose Python for Lambda due to simplicity and familiarity.
* Used DynamoDB for scalability and easy integration with Lambda.
* Selected AWS SAM for IaaC for its ease of use with AWS services.

**Challenges Faced:**

* Mapping phone numbers to vanity numbers.
* Ensuring real-time interaction with Amazon Connect.

**Shortcuts Taken:**

* Simplified the "best" vanity number selection.
* Limited error handling for simplicity.
* What Would Be Done With More Time:
* Implement more robust error handling.

Optimize vanity number generation logic.

**Enhance security measures.**

* Considerations for High Traffic and Security:
* Implement rate limiting.
* Use AWS WAF for protection against attacks.
* Optimize DynamoDB read/write capacity based on traffic.

**Architecture Diagram**

Caller

DynamoDB

Vanity Numbers Table

AWS Lambda

Vanity Number Function

Amazon Connect

Contact Flow Trigger