**Prerequisites & Setup**

**Step 1: Create AWS Account**

* Sign up for an AWS account if you don't have one
* You'll need a credit card, but SMS costs are very low (around $0.0075 per message in the US)

**Step 2: Set Up User Permissions**

* Go to AWS IAM (Identity and Access Management)
* Create a new user specifically for your app
* Give this user only SNS permissions (principle of least privilege)
* Generate access keys (Access Key ID and Secret Access Key) - keep these secure!

**Configure SNS Service**

**Step 3: Set Up SNS**

* Navigate to AWS SNS in the console
* Choose your preferred AWS region (closer to your users = faster delivery)
* No initial setup required - SNS is ready to use immediately

**Step 4: Request Production Access (Important!)**

* By default, AWS puts you in "SMS Sandbox mode"
* In sandbox, you can only send to verified phone numbers
* Submit a support case to request production SMS access
* This usually takes 24-48 hours and requires explaining your use case

**Mobile App Integration**

**Step 5: Install AWS SDK**

* **iOS**: Add AWS SDK via CocoaPods, SPM, or Carthage
* **Android**: Add AWS SDK via Gradle
* **React Native/Flutter**: Use respective AWS plugins

**Step 6: Configure AWS Credentials in App**

* Store your access keys securely (use environment variables, not hardcoded)
* Initialize AWS SNS client with your credentials and region
* Consider using AWS Cognito for more secure credential management

**Implement Text Messaging**

**Step 7: Get User Phone Numbers**

* Add phone number input in your app's UI
* Format numbers in international format (+1234567890)
* Implement validation to ensure proper format

**Step 8: Handle Opt-in Requirements**

* **Legal requirement**: Users must explicitly opt-in to receive SMS
* Add clear consent checkboxes/buttons
* Store opt-in status in your database
* Provide easy opt-out instructions in messages

**Step 9: Send Messages Programmatically**

Basic process:

1. User triggers action requiring SMS (signup, verification, etc.)

2. Your app calls AWS SNS API

3. Provide: phone number, message text, message type

4. AWS delivers the message

5. Handle the response (success/failure)

**Step 10: Implement Message Types**

* **Transactional**: Order confirmations, account alerts (higher priority)
* **Promotional**: Marketing messages (lower priority, more restrictions)
* Choose appropriate type when sending

**Error Handling & Best Practices**

**Step 11: Handle Common Issues**

* Invalid phone numbers
* Carrier blocking
* Rate limiting (1 message per second by default)
* Message delivery failures
* Implement retry logic for failed messages

**Step 12: Monitor and Track**

* Set up CloudWatch monitoring for delivery rates
* Track costs in AWS billing dashboard
* Monitor for spam complaints or blocks

**Testing Strategy**

**Step 13: Test Thoroughly**

* Start with sandbox mode using verified numbers
* Test different phone carriers and international numbers
* Test message length limits (160 characters for single SMS)
* Test special characters and emojis

**Step 14: Production Deployment**

* Once production access approved, test with real users gradually
* Monitor delivery rates and user feedback
* Have a plan to handle high-volume sending if needed

**Ongoing Maintenance**

**Step 15: Compliance and Monitoring**

* Respect opt-out requests immediately
* Keep records of consent
* Monitor for delivery issues
* Stay updated on carrier policy changes
* Regularly review AWS costs