

SAHAYA – Disaster Response Management System

Phase 7 & 8: Integration & Data Management

Phase 7: Integration & External Access

1. Introduction:

The SAHAYA system is architecturally prepared for future integration with external emergency services and data sources. While current implementation focuses on core functionality, the framework supports seamless expansion.

2. Integration Architecture:

- API-Based Integration: REST API endpoints ready for weather alerts and government disaster feeds
- Authentication: Named Credentials configured for secure external system access
- Real-time Communication: Platform Events framework for instant notifications

3. Planned External Integrations:

- Weather Service API: Real-time severe weather alerts and disaster warnings
- SMS/Email Gateways: Bulk emergency notifications to volunteers and affected communities
- Government Disaster Feeds: Official disaster declarations and resource allocation data
- Hospital Systems: Patient transfer coordination and medical record exchange

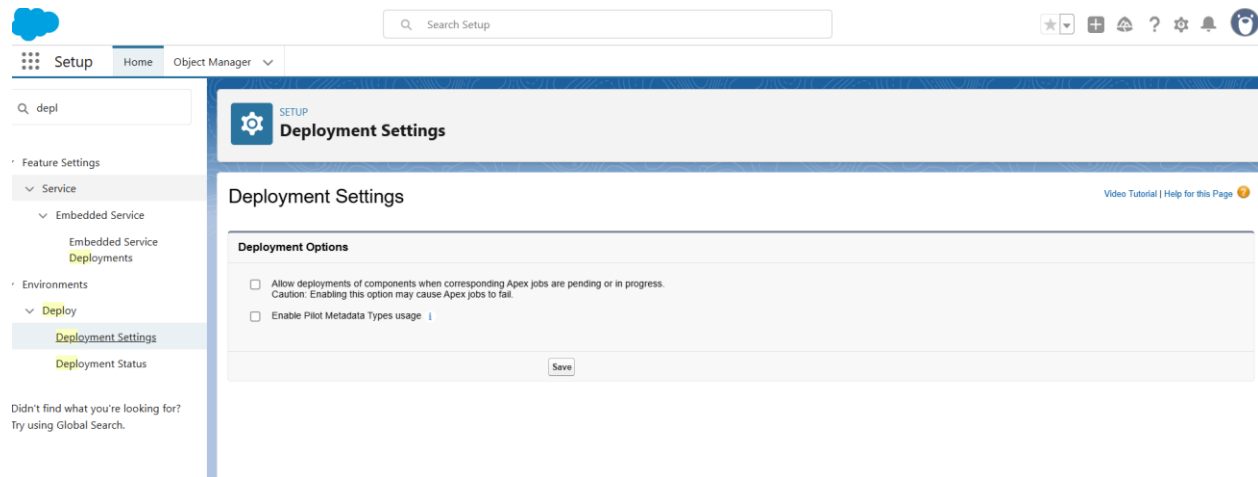
4. Technical Implementation Readiness:

- Named Credentials: Secure authentication setup for external services
- REST/SOAP Capabilities: Apex HTTP callout methods implemented
- Error Handling: Robust exception management for external system failures

Phase 8: Data Management & Deployment

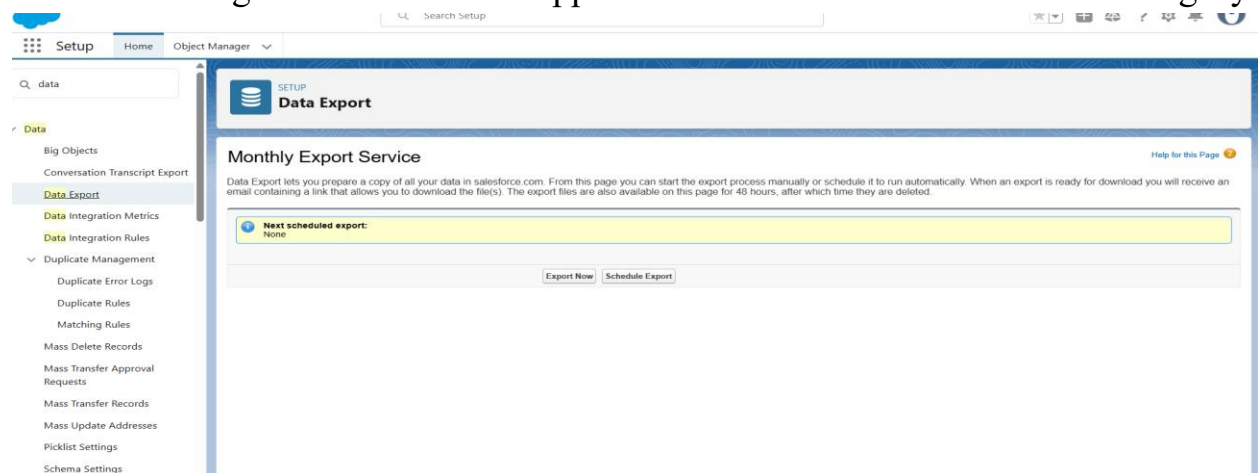
1. Deployment Strategy:

- Development Environment: Salesforce Developer Edition with full development capabilities
- Production Deployment: Change Sets for controlled component migration
- Version Control: Salesforce DX integration for source-driven development



2. Data Management Framework

- Initial Data Load: Data Loader templates for disaster records, medical camps, and volunteer data
- Regular Backups: Weekly export schedules for critical operational data
- Data Migration: CSV-based approach with validation rules for data integrity



3. Deployment Components Custom Objects:

Disaster__c, Medical_Camp__c, Patient__c, Volunteer_Assignment__c, Resource__c

Apex Components:

- VolunteerService.cls (100% test coverage)
- MedicalCampTrigger.trigger, DisasterTrigger.trigger

Lightning Experience:

- Custom Lightning App: "Disaster Response"
- Optimized Record Pages for all core objects
- Command Center Dashboard for management

The screenshot shows the Salesforce Setup interface for Apex Classes. The left sidebar contains navigation links for Email, Custom Code, Apex Classes, Apex Settings, Apex Test Execution, Apex Test History, Apex Triggers, Environments, Jobs, Apex Flex Queue, and Apex Jobs. The main content area displays the 'Apex Classes' page with a warning banner about code coverage. Below the banner, there is a table listing Apex classes with columns for Action, Name, Namespace Prefix, Api Version, Status, Size Without Comments, Last Modified By, and Has Trace Flags. The table lists several classes, including DeveloperEditionUtils, DeveloperEditionUtilsTest, PostInstallScript, PostInstallScriptTest, TestVolunteerService, and VolunteerService. Below the table, there is a section for 'Dynamic Apex Classes' with a similar table structure, but it shows 'No records to display'.

Action	Name	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit Security	DeveloperEditionUtils	devedapp	64.0	Active	164	OrgFarm.EPIC, 9/19/2025, 9:05 AM	<input type="checkbox"/>
Edit	DeveloperEditionUtilsTest	devedapp	64.0	Active	261	OrgFarm.EPIC, 9/19/2025, 9:05 AM	<input type="checkbox"/>
Edit Security	PostInstallScript	devedapp	64.0	Active	2,175	OrgFarm.EPIC, 9/19/2025, 9:05 AM	<input type="checkbox"/>
Edit	PostInstallScriptTest	devedapp	64.0	Active	781	OrgFarm.EPIC, 9/19/2025, 9:05 AM	<input type="checkbox"/>
Edit Del	TestVolunteerService		64.0	Active	1,598	Tejasri.Vasamsetti, 9/25/2025, 3:15 PM	<input type="checkbox"/>
Edit Del Security	VolunteerService		64.0	Active	2,194	Tejasri.Vasamsetti, 9/25/2025, 3:01 PM	<input type="checkbox"/>

The screenshot shows the Salesforce Setup interface for Apex Triggers. The left sidebar contains navigation links for Email, Custom Code, Apex Classes, Apex Settings, Apex Test Execution, Apex Test History, Apex Triggers, Environments, Jobs, Apex Flex Queue, and Apex Jobs. The main content area displays the 'Apex Triggers' page with a warning banner about code coverage. Below the banner, there is a table listing Apex triggers with columns for Action, Name, Namespace Prefix, sObject Type, Api Version, Status, Size Without Comments, Last Modified By, and Has Trace Flags. The table lists two triggers: DisasterTrigger and MedicalCampTrigger. Below the table, there is a section for 'Dynamic Apex Classes' with a similar table structure, but it shows 'No records to display'.

Action	Name	Namespace Prefix	sObject Type	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit Del	DisasterTrigger		Disaster	64.0	Active	183	Tejasri.Vasamsetti, 9/25/2025, 3:02 PM	<input type="checkbox"/>
Edit Del	MedicalCampTrigger		Medical_Camp	64.0	Active	234	Tejasri.Vasamsetti, 9/25/2025, 2:51 PM	<input type="checkbox"/>

4. Backup & Recovery:

- Automated Exports: Scheduled weekly data exports

- Selective Backup: Priority on disaster records, patient data, and resource inventory
- Recovery Plan: Step-by-step data restoration procedures

5. Change Management :

- Component Packaging: Outbound change sets containing all customizations
- Dependency Management: Automated validation of component relationships
- Deployment Testing: Sandbox validation before production deployment

6. Conclusion:

Phases 7 and 8 establish SAHAYA as a production-ready system with robust data management capabilities and clear integration pathways for future expansion. The deployment framework ensures smooth transition to operational environments while maintaining data integrity and system reliability.