VAST As-Built Report Generator

Project Resources

Topic	Content
1.0 - Concept	Initial Concept - As-Built Report Generator
2.0 - PRD	Project Requirements Document
3.0 - Project Plan	Project Plan, Phases, & Timeline
4.0 - Tasks	Initial Development Tasks
5.0 - Status	Development Status
6.0 - Design	Design Guide
7.0 - AI Guardrails	AI Development Reference Guide
8.0 - Installation	Installation Procedure
9.0 - Report	Report Example (Mock Data)
10.0 - API Reference	VAST API v7 Data Gathering Analysis

Github Repository (develop) - ps-deploy-report

VAST_As-Built_Report_Generator.pdf

VAST As-Built Report Generator

A Python command-line tool that automatically generates comprehensive, professionally formatted "as-built" reports for VAST Data clusters following deployment by Professional Services.

Overview

This tool connects to a VAST Data cluster via its REST API, extracts configuration and status information, and generates both human-readable PDF reports and machine-readable JSON data files. It is designed to streamline the post-deployment documentation process for VAST Professional Services engineers.

Features

- **Automated Data Collection**: Connects to VAST clusters and extracts comprehensive configuration data
- **Dual Output Formats**: Generates both PDF reports for customers and JSON files for automation
- **Professional Formatting**: Creates customer-ready PDF documents with proper styling and organization
- **Fault Tolerance**: Handles network failures and missing data gracefully
- **Secure Authentication**: Supports secure credential handling without storing sensitive data
- **Comprehensive Logging**: Detailed logging for troubleshooting and audit purposes

Requirements

- Python 3.8 or higher
- Network access to VAST Management Service (VMS)
- Valid VAST cluster credentials with read access
- Dependencies listed in `requirements.txt`

PENDING:

Installation

1. Clone the repository:

```
git clone <https://github.com/rstamps01/ps-deploy-report.git>
cd ps-deploy-report
```

2. Create and activate a virtual environment:

```
python3 -m venv venv
source venv/bin/activate  # On Windows: venv\\Scripts\\activate
```

3. Install dependencies:

```
pip install -r requirements.txt
```

Configuration

1. Copy the configuration template:

```
cp config/config.yaml.template config/config.yaml
```

Edit `config/config.yaml` with your environment-specific settings

Usage

Basic usage:

```
python src/main.py --cluster-ip 192.168.1.100 --output-dir ./output
```

The tool will prompt for credentials securely at runtime.

Output

The tool generates two types of output:

1. **PDF Report**: Professional customer-facing document
   a. (`output/vast_report_YYYYMMDD_HHMMSS.pdf`)
2. **JSON Data**: Machine-readable data file:
   a. (`output/vast_data_YYYYMMDD_HHMMSS.json`)

Project Structure

```
ps-deploy-report/
├── README.md                    # This file
├── STATUS.md                    # Development status tracking
├── requirements.txt             # Python dependencies
├── config/                      # Configuration files
├── src/                         # Source code
├── tests/                       # Unit and integration tests
├── templates/                   # Report templates
├── logs/                        # Application logs
└── output/                      # Generated reports
```

Development

This project follows the development guidelines outlined in the AI Development Reference Guide. See `STATUS.md` for current development status and next steps.

Support

For issues, questions, or contributions, please refer to the project's GitHub repository.

License

[License information to be added]

Version: 1.0.0-dev
Target VAST Version: 5.3
API Version: 7