# **SD-WAN Configuration List Playbook Documentation**

## **Overview**

The SD-WAN Configuration List playbook is an Ansible automation script designed to inventory and list all configuration items from Cisco SD-WAN environments. This playbook uses the Sastre tool to query the vManage controller and extract detailed information about templates, policies, and configuration groups, creating organized reports for documentation and auditing purposes.

## **Purpose**

This playbook serves several key functions:

* **Configuration Inventory**: Creates a comprehensive inventory of all SD-WAN configuration elements
* **Documentation**: Generates structured reports for compliance and documentation purposes
* **Auditing**: Provides snapshot of current configuration state for change management
* **Troubleshooting**: Assists in identifying available templates and policies during issue resolution
* **Migration Planning**: Helps assess configuration complexity before migrations or upgrades

## **Prerequisites**

### **Software Requirements**

* **Ansible**: Version 2.9 or higher
* **Sastre**: Cisco SD-WAN Automation Toolset installed and accessible in PATH
* **Python**: Version 3.6 or higher (required by Sastre)

### **Network Requirements**

* **Connectivity**: Network access to vManage controller on HTTPS port (default 443)
* **Authentication**: Valid vManage user credentials with read permissions
* **Firewall**: Ensure firewall rules allow HTTPS traffic to vManage

### **Access Requirements**

* **vManage Access**: User account with minimum read-only permissions
* **API Access**: vManage API must be accessible and enabled
* **Certificate**: SSL certificate validation can be bypassed (playbook includes this setting)

## **Playbook Structure**

### **Variables Configuration**

vars:

# Use environment variables for credentials

vmanage\_host: "{{ lookup('env', 'VMANAGE\_HOST') | default('') }}"

vmanage\_username: "{{ lookup('env', 'VMANAGE\_USERNAME') | default('') }}"

vmanage\_password: "{{ lookup('env', 'VMANAGE\_PASSWORD') | default('') }}"

vmanage\_port: "{{ lookup('env', 'VMANAGE\_PORT') | default('443') }}"

# Directory structure

generated\_dir: "{{ playbook\_dir }}/generated"

list\_timestamp: "{{ ansible\_date\_time.epoch }}"

**Security Note**: All sensitive credentials are sourced from environment variables to maintain security best practices.

### **Environment Variables**

| **Variable** | **Required** | **Default** | **Description** |
| --- | --- | --- | --- |
| VMANAGE\_HOST | Yes | None | vManage controller hostname or IP address |
| VMANAGE\_USERNAME | Yes | None | vManage username with read permissions |
| VMANAGE\_PASSWORD | Yes | None | Password for vManage authentication |
| VMANAGE\_PORT | No | 443 | HTTPS port for vManage access |

### **Directory Structure**

The playbook creates the following directory structure:

playbook\_directory/

└── generated/

├── device\_template\_list\_<timestamp>.txt

├── feature\_template\_list\_<timestamp>.txt

├── policy\_definition\_list\_<timestamp>.txt

├── policy\_list\_<timestamp>.txt

├── configuration\_group\_list\_<timestamp>.txt

└── consolidated\_inventory\_<timestamp>.txt

## **Detailed Task Analysis**

### **Task 1: Environment Variable Validation**

- name: Validate environment variables are set

**Purpose**: Ensures all required credentials are available before proceeding

**Function**:

* Validates that vmanage\_host, vmanage\_username, and vmanage\_password are set
* Fails immediately if any critical environment variables are missing
* Prevents failed execution due to missing authentication parameters

**Error Handling**: Stops playbook execution with clear error message indicating which variable is missing

### **Task 2: Directory Creation**

- name: Create generated directory

**Purpose**: Creates the output directory structure for storing inventory files

**Function**:

* Creates generated directory under playbook location
* Sets appropriate permissions (755) for directory access
* Ensures consistent file organization

**Output**: Creates directory structure for storing all generated reports

### **Task 3: vManage Connectivity Test**

- name: Test vManage connectivity

**Purpose**: Verifies vManage controller accessibility before attempting configuration queries

**Function**:

* Makes REST API call to /dataservice/system/device/controllers endpoint
* Uses basic authentication with provided credentials
* Sets 60-second timeout for connection attempts
* Bypasses SSL certificate validation for internal deployments

**Parameters**:

* **URL**: https://{{ vmanage\_host }}/dataservice/system/device/controllers
* **Method**: GET
* **Authentication**: Basic Auth
* **Timeout**: 60 seconds
* **SSL Validation**: Disabled

**Success Criteria**: HTTP 200 response code indicates successful connectivity

### **Task 4: Device Templates Listing**

- name: List Device Templates

**Purpose**: Retrieves comprehensive list of all device templates from vManage

**Sastre Command**:

sastre --address <vmanage\_host> --port <port> --user <username> --password <password> list configuration template\_device

**Generated Content**:

* Complete inventory of device templates
* Template names, types, and associated information
* Device template metadata and configuration details

**Output File**: device\_template\_list\_<timestamp>.txt

### **Task 5: Feature Templates Listing**

- name: List Feature Templates

**Purpose**: Catalogs all feature templates available in the SD-WAN environment

**Sastre Command**:

sastre --address <vmanage\_host> --port <port> --user <username> --password <password> list configuration template\_feature

**Generated Content**:

* Inventory of all feature templates (System, VPN, Interface, Routing, Security, etc.)
* Template names and associated device types
* Feature template categories and descriptions

**Output File**: feature\_template\_list\_<timestamp>.txt

### **Task 6: Policy Definitions Listing**

- name: List Policy Definitions

**Purpose**: Documents all policy definitions configured in the SD-WAN fabric

**Sastre Command**:

sastre --address <vmanage\_host> --port <port> --user <username> --password <password> list configuration policy\_definition

**Generated Content**:

* Complete list of policy definitions
* Policy types (QoS, Security, Traffic Engineering, etc.)
* Policy definition names and categories

**Output File**: policy\_definition\_list\_<timestamp>.txt

### **Task 7: Policy Lists Inventory**

- name: List Policy Lists

**Purpose**: Captures all policy lists used for traffic classification and filtering

**Sastre Command**:

sastre --address <vmanage\_host> --port <port> --user <username> --password <password> list configuration policy\_list

**Generated Content**:

* Data prefix lists, application lists, color lists, etc.
* Policy list names and types
* Associated policy list entries and values

**Output File**: policy\_list\_<timestamp>.txt

### **Task 8: Configuration Groups Listing**

- name: List Configuration Groups

**Purpose**: Documents configuration groups (if available in the environment)

**Sastre Command**:

sastre --address <vmanage\_host> --port <port> --user <username> --password <password> list configuration --tags all

**Generated Content**:

* Configuration group names and associated devices
* Group deployment status and configurations
* Hierarchical configuration management information

**Output File**: configuration\_group\_list\_<timestamp>.txt

**Error Handling**: Uses ignore\_errors: true as configuration groups may not be available in all SD-WAN deployments

### **Task 9: Consolidated Report Generation**

- name: Create consolidated inventory file

**Purpose**: Creates a comprehensive, single-file report combining all inventory data

**Generated Content**:

* **Header Information**: Timestamp, vManage host, generation details
* **Device Templates Section**: Complete device template inventory
* **Feature Templates Section**: All feature template listings
* **Policy Definitions Section**: Policy definition catalog
* **Policy Lists Section**: Policy list inventory
* **Configuration Groups Section**: Configuration group information (if available)

**Output File**: consolidated\_inventory\_<timestamp>.txt

**Report Structure**:

SD-WAN Configuration Inventory

Generated: 2025-08-16T23:10:32.123456Z

vManage: vmanage.company.com

=== DEVICE TEMPLATES ===

[Device template listings]

=== FEATURE TEMPLATES ===

[Feature template listings]

=== POLICY DEFINITIONS ===

[Policy definition listings]

=== POLICY LISTS ===

[Policy list listings]

=== CONFIGURATION GROUPS ===

[Configuration group listings or 'Not available']

## **Usage Instructions**

### **Manual Execution**

**Set Environment Variables**:  
  
 export VMANAGE\_HOST="vmanage.company.com"

export VMANAGE\_USERNAME="automation\_user"

export VMANAGE\_PASSWORD="secure\_password"

export VMANAGE\_PORT="443" # Optional, defaults to 443

**Execute Playbook**:  
  
 ansible-playbook sdwan\_list.yml

**Verify Output**:  
  
 ls -la generated/

cat generated/consolidated\_inventory\_<timestamp>.txt

### **Pipeline Integration**

For GitLab CI/CD integration, create a pipeline job:

list-sdwan-config:

stage: inventory

script:

- ansible-playbook sdwan\_list.yml

artifacts:

paths:

- generated/

expire\_in: 30 days

variables:

VMANAGE\_HOST: "${VMANAGE\_HOST}"

VMANAGE\_USERNAME: "${VMANAGE\_USERNAME}"

VMANAGE\_PASSWORD: "${VMANAGE\_PASSWORD}"

### **Scheduled Execution**

For regular inventory updates, schedule the playbook using cron:

# Daily inventory at 2 AM

0 2 \* \* \* cd /path/to/playbook && ansible-playbook sdwan\_list.yml

## **Output Files Description**

### **Individual Component Files**

1. **device\_template\_list\_<timestamp>.txt**
   * Contains: Device template inventory
   * Format: Tabular listing with template names, types, and details
   * Use Case: Device template management and documentation
2. **feature\_template\_list\_<timestamp>.txt**
   * Contains: Feature template catalog
   * Format: Organized by feature type and template name
   * Use Case: Template reuse analysis and feature configuration auditing
3. **policy\_definition\_list\_<timestamp>.txt**
   * Contains: Policy definition inventory
   * Format: Policy names, types, and definitions
   * Use Case: Policy compliance and security auditing
4. **policy\_list\_<timestamp>.txt**
   * Contains: Policy list inventory
   * Format: List names, types, and entries
   * Use Case: Traffic classification and filtering documentation
5. **configuration\_group\_list\_<timestamp>.txt** (optional)  
   * Contains: Configuration group information
   * Format: Group names, associated devices, status
   * Use Case: Centralized configuration management documentation

### **Consolidated Report**

**consolidated\_inventory\_<timestamp>.txt**

* **Purpose**: Single-file comprehensive inventory
* **Format**: Structured sections with clear delimiters
* **Benefits**:
  + Easy sharing and documentation
  + Complete snapshot of configuration state
  + Suitable for compliance reporting
  + Historical comparison capabilities

## **Error Handling and Troubleshooting**

### **Common Issues and Solutions**

**Environment Variable Not Set** Error: Required environment variable VMANAGE\_HOST is not set

1. **Solution**: Ensure all required environment variables are properly exported

**Connection Timeout** Error: Cannot connect to vManage at <host>

1. **Solutions**:  
   * Verify network connectivity to vManage
   * Check firewall rules and port accessibility
   * Confirm vManage hostname/IP address
   * Validate vManage service status

**Authentication Failure** Error: HTTP 401 Unauthorized

1. **Solutions**:  
   * Verify username and password credentials
   * Check user account status and permissions
   * Ensure user has API access enabled

**Sastre Command Not Found** Error: sastre: command not found

1. **Solutions**:  
   * Install Sastre using pip install sdwan
   * Verify Sastre is in system PATH
   * Check Python environment and dependencies

**SSL Certificate Issues** Error: SSL certificate verification failed

1. **Solution**: Playbook includes SSL bypass settings, but verify network security policies

### **Debug Mode**

Enable verbose output for troubleshooting:

ansible-playbook sdwan\_list.yml -v # Basic verbosity

ansible-playbook sdwan\_list.yml -vv # More detailed output

ansible-playbook sdwan\_list.yml -vvv # Debug level output

### **Log Analysis**

Check Sastre environment variables and output:

# Verify environment variables

echo $VMANAGE\_HOST

echo $VMANAGE\_USERNAME

echo $VMANAGE\_PORT

# Test Sastre connectivity manually

sastre --address $VMANAGE\_HOST --user $VMANAGE\_USERNAME --password $VMANAGE\_PASSWORD list configuration template\_device

## **Security Considerations**

### **Credential Management**

* **Environment Variables**: Use environment variables for all sensitive data
* **Vault Integration**: Consider Ansible Vault for additional security
* **Access Control**: Limit playbook access to authorized personnel
* **Audit Logging**: Enable logging for compliance and security auditing

### **Network Security**

* **SSL/TLS**: Playbook bypasses SSL validation for internal certificates
* **Firewall Rules**: Ensure minimal required network access
* **VPN Access**: Consider running from secure management networks
* **Network Segmentation**: Isolate management traffic appropriately

### **Data Protection**

* **Output Security**: Protect generated inventory files appropriately
* **Retention Policy**: Implement appropriate data retention policies
* **Access Controls**: Secure generated reports with proper file permissions
* **Backup Encryption**: Encrypt sensitive configuration data in backups

## **Integration and Automation**

### **CI/CD Pipeline Integration**

The playbook integrates seamlessly with CI/CD pipelines for:

* **Scheduled Inventories**: Regular configuration documentation
* **Change Management**: Pre/post-change configuration snapshots
* **Compliance Reporting**: Automated compliance documentation
* **Audit Preparation**: Systematic configuration documentation

### **API Integration**

Sastre output can be integrated with:

* **Configuration Management Systems**: Import into CMDB systems
* **Monitoring Platforms**: Feed into network monitoring tools
* **Documentation Systems**: Automated documentation generation
* **Change Management**: Integration with change tracking systems

### **Reporting and Analytics**

Generated data supports:

* **Trend Analysis**: Historical configuration growth and changes
* **Capacity Planning**: Template and policy utilization analysis
* **Compliance Monitoring**: Regular compliance state assessment
* **Risk Assessment**: Configuration complexity and security analysis

## **Best Practices**

### **Execution Practices**

1. **Regular Scheduling**: Run inventory collection on regular schedule
2. **Version Control**: Store playbooks in version control systems
3. **Testing**: Test in lab environment before production execution
4. **Documentation**: Maintain execution logs and results documentation

### **Security Practices**

1. **Credential Rotation**: Regularly rotate service account credentials
2. **Access Auditing**: Monitor and audit playbook execution
3. **Network Security**: Use secure networks for execution
4. **Data Classification**: Classify and protect generated inventory data

### **Operational Practices**

1. **Backup Management**: Maintain backups of generated inventory data
2. **Change Tracking**: Track changes in configuration inventories over time
3. **Integration Testing**: Test integration with downstream systems
4. **Performance Monitoring**: Monitor execution time and resource usage

## **Maintenance and Updates**

### **Regular Maintenance Tasks**

* **Sastre Updates**: Keep Sastre tool updated to latest version
* **Ansible Updates**: Maintain current Ansible version
* **Credential Updates**: Update authentication credentials as needed
* **Network Updates**: Adjust for network infrastructure changes

### **Monitoring and Alerting**

* **Execution Monitoring**: Monitor playbook execution success/failure
* **Output Validation**: Validate generated inventory content
* **Performance Tracking**: Track execution time and resource usage
* **Error Alerting**: Configure alerts for execution failures

This playbook provides a comprehensive, secure, and automated approach to SD-WAN configuration inventory management, supporting both operational needs and compliance requirements.