# **SD-WAN Show-Template Playbook Documentation**

## **Overview**

The sastre\_show\_template.yml playbook is an Ansible automation script designed to retrieve and display template details from Cisco SD-WAN environments. This playbook uses the Sastre tool to extract template information from the vManage controller and saves the results in a structured JSON format.

## **Use Case**

**Use Case #3**: sastre show-template - Display template details

This playbook addresses the need to:

* View all feature templates configured in the SD-WAN environment
* View all device templates configured in the SD-WAN environment
* Export template details for documentation or analysis purposes
* Provide a simple way to inventory templates without complex reporting

## **Prerequisites**

### **Software Requirements**

* Ansible 2.9 or higher
* Sastre tool installed and accessible in PATH
* Python 3.6 or higher
* Network connectivity to vManage controller

### **Environment Variables**

The following environment variables must be set before running the playbook:

| **Variable** | **Description** | **Example** |
| --- | --- | --- |
| VMANAGE\_HOST | vManage controller hostname/IP | vmanage-prod.company.com |
| VMANAGE\_USERNAME | Username for vManage authentication | automation |
| VMANAGE\_PASSWORD | Password for vManage authentication | SecurePassword123 |

### **vManage Compatibility**

* Tested with vManage version 20.15
* Compatible with HTTPS (port 443)
* Requires API access permissions for the specified user

## **Playbook Structure**

### **Variables Configuration**

vars:

vmanage\_host: "{{ lookup('env', 'VMANAGE\_HOST') | default('vmanage-amfament-prod.sdwan.cisco.com') }}"

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

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

vmanage\_port: "443"

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

timestamp: "{{ ansible\_date\_time.epoch }}"

output\_file: "{{ generated\_dir }}/template\_details\_{{ timestamp }}.json"

### **Directory Structure**

The playbook creates a simple directory structure:

playbook\_directory/

├── sastre\_show\_template.yml

└── generated/

└── template\_details\_[timestamp].json

## **Task Analysis**

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

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

**What it does**:

* Validates that VMANAGE\_HOST, VMANAGE\_USERNAME, and VMANAGE\_PASSWORD are set
* Fails immediately if any required environment variable is missing
* Prevents failed execution due to missing credentials

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

**Purpose**: Creates the output directory for results

**What it does**:

* Creates the generated directory in the same location as the playbook
* Sets appropriate permissions (755)
* Ensures the output location exists before execution

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

**Purpose**: Verifies the vManage controller is accessible before attempting operations

**What it does**:

* Makes a REST API call to /dataservice/system/device/controllers
* Uses basic authentication with provided credentials
* Sets 60-second timeout
* Ignores SSL certificate validation for internal certificates
* Stores results for validation

### **Task 4: Connectivity Validation**

**Purpose**: Stops execution if connectivity test fails

**What it does**:

* Checks if the connectivity test returned HTTP 200
* Fails the playbook if vManage is unreachable
* Prevents unnecessary Sastre commands when connectivity issues exist

### **Task 5: Feature Templates Retrieval**

**Purpose**: Retrieves all feature templates from vManage

**Sastre command executed**:

sastre --address [host] --port [port] --user [username] --password [password] --verbose show-template --workdir [generated\_dir] feature

**What it does**:

* Connects to vManage using provided credentials
* Retrieves all feature template details
* Uses verbose output for detailed information
* Stores output in the feature\_templates variable

### **Task 6: Device Templates Retrieval**

**Purpose**: Retrieves all device templates from vManage

**Sastre command executed**:

sastre --address [host] --port [port] --user [username] --password [password] --verbose show-template --workdir [generated\_dir] device

**What it does**:

* Connects to vManage using provided credentials
* Retrieves all device template details
* Uses verbose output for detailed information
* Stores output in the device\_templates variable

### **Task 7: Results Export**

**Purpose**: Saves template details to a structured JSON file

**Generated file structure**:

{

"timestamp": "2025-08-17T12:34:56Z",

"vmanage\_host": "vmanage-prod.company.com",

"feature\_templates": "[template details output]",

"device\_templates": "[template details output]"

}

**What it does**:

* Creates a timestamped JSON file
* Includes execution timestamp and vManage host for reference
* Stores both feature and device template details
* Saves file in the generated directory

### **Task 8: Completion Notification**

**Purpose**: Provides execution status and file location

**What it displays**:

* Path to the generated output file
* Return codes for both template retrieval commands
* Success/failure indication

## **Output Format**

### **File Naming Convention**

Output files follow the pattern: template\_details\_[unix\_timestamp].json

Example: template\_details\_1692274496.json

### **JSON Structure**

The output JSON contains four main elements:

* **timestamp**: ISO 8601 formatted execution time
* **vmanage\_host**: The vManage controller that was queried
* **feature\_templates**: Complete output from the feature template query
* **device\_templates**: Complete output from the device template query

## **Execution Methods**

### **Manual Execution**

# Set environment variables

export VMANAGE\_HOST="your-vmanage-host.com"

export VMANAGE\_USERNAME="your-username"

export VMANAGE\_PASSWORD="your-password"

# Run the playbook

ansible-playbook sastre\_show\_template.yml

### **Pipeline Integration**

The playbook can be integrated into GitLab CI/CD pipelines:

show-templates:

stage: sdwan-operations

script:

- ansible-playbook sastre\_show\_template.yml

artifacts:

paths:

- generated/

expire\_in: 7 days

variables:

VMANAGE\_HOST: $VMANAGE\_HOST

VMANAGE\_USERNAME: $VMANAGE\_USERNAME

VMANAGE\_PASSWORD: $VMANAGE\_PASSWORD

## **Security Considerations**

### **Credential Protection**

* Environment variables prevent hardcoded credentials
* Passwords are never displayed in debug output
* SSL certificate validation is disabled for internal certificates only

### **Access Requirements**

The specified user account must have:

* Read access to template configurations
* API access permissions in vManage
* Network connectivity to the vManage controller

## **Troubleshooting**

### **Common Issues**

**Environment Variable Not Set**

TASK [Validate environment variables are set] \*\*\*\*\*

fatal: [localhost]: FAILED! => {"msg": "Required environment variable VMANAGE\_HOST is not set"}

**Solution**: Ensure all required environment variables are exported before running

**Connectivity Failure**

TASK [Fail if connectivity test failed] \*\*\*\*\*\*\*\*\*\*\*

fatal: [localhost]: FAILED! => {"msg": "Cannot connect to vManage at vmanage-host.com"}

**Solution**: Verify network connectivity, credentials, and vManage availability

**Sastre Command Not Found**

TASK [Show all feature templates] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fatal: [localhost]: FAILED! => {"msg": "sastre: command not found"}

**Solution**: Install Sastre tool and ensure it's in the system PATH

### **Success Indicators**

* All tasks complete without errors
* JSON file created in the generated directory
* Return codes of 0 for both template queries
* File size greater than 0 bytes

## **Maintenance**

### **Regular Tasks**

* Monitor playbook execution for failures
* Verify output files contain expected data
* Update credentials when they rotate
* Test connectivity after vManage updates

### **File Management**

* Archive old template detail files as needed
* Monitor disk space usage in the generated directory
* Implement retention policies based on organizational requirements

## **Integration Notes**

This playbook is designed to work as part of a larger SD-WAN automation suite and can be:

* Called from other playbooks for template inventory
* Integrated into scheduled jobs for regular template auditing
* Used as input for configuration management workflows
* Combined with other Sastre operations for comprehensive SD-WAN management