# **SD-WAN List All Devices Playbook Documentation**

## **Overview**

The **list\_all\_devices.yml** playbook is an Ansible automation script designed to retrieve and document all devices registered in a Cisco SD-WAN vManage controller. This playbook connects to vManage via REST API, retrieves comprehensive device inventory information, and exports the data in multiple formats for analysis and reporting purposes.

## **Use Case**

**Use Case 9: List all devices - Get inventory of all devices**

This playbook addresses the need to:

* Retrieve complete inventory of all SD-WAN devices from vManage
* Document device details including model, status, and connectivity information
* Export device data in JSON and CSV formats for analysis
* Monitor device reachability and operational status
* Create device inventory reports for asset management
* Provide automated device discovery for documentation updates

## **Prerequisites**

### **Environment Variables**

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

| **Variable** | **Description** | **Default Value** |
| --- | --- | --- |
| **VMANAGE\_HOST** | vManage controller hostname/IP | vmanage-amfament-prod.sdwan.cisco.com |
| **VMANAGE\_USERNAME** | Username for vManage authentication | automation |
| **VMANAGE\_PASSWORD** | Password for vManage authentication |  |

## **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"

### **Directory Structure**

The playbook creates the following directory structure:

playbook\_directory/

├── list\_all\_devices.yml

└── ../generated/

├── all\_devices\_complete.json

├── all\_devices\_simplified.json

└── all\_devices.csv

## **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 execution failures due to missing credentials
* Provides clear error messages for troubleshooting

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

**Purpose:** Creates the output directory for generated device reports

**What it does:**

* Creates the **generated** directory relative to the playbook location
* Sets appropriate permissions (755) for file access
* Ensures the output location exists before data export
* Creates parent directories if they don't exist

#### **Task 3: vManage Authentication**

**Purpose:** Establishes authenticated session with vManage controller

**What it does:**

* Makes a POST request to **/j\_security\_check** endpoint
* Sends credentials as form-urlencoded data
* Validates successful authentication (HTTP 200 status)
* Stores session cookie for subsequent API calls
* Handles SSL certificate validation for HTTPS connections

#### **Task 4: Session Token Retrieval**

**Purpose:** Obtains XSRF token for API security

**What it does:**

* Makes a GET request to **/dataservice/client/token** endpoint
* Uses session cookie from authentication step
* Retrieves XSRF token for API request protection
* Stores token for use in subsequent API calls

#### **Task 5: Token Extraction**

**Purpose:** Processes and stores the security token

**What it does:**

* Extracts token value from response content or body
* Handles different token response formats
* Sets token as fact for use in API headers
* Provides fallback for empty token scenarios

#### **Task 6: Device Inventory Retrieval**

**Purpose:** Fetches complete device inventory from vManage

**API endpoint:** **/dataservice/device**

**What it does:**

* Makes authenticated GET request to device API endpoint
* Includes session cookie and XSRF token in headers
* Sets **60-second timeout** for large inventories
* Retrieves all registered devices in JSON format
* Handles API response with error checking

#### **Task 7: Device Data Parsing**

**Purpose:** Processes raw device data from API response

**What it does:**

* Extracts device array from JSON response
* Handles empty or missing data gracefully
* Stores device list as Ansible fact
* Prepares data for further processing

#### **Task 8: Device Information Extraction**

**Purpose:** Creates simplified device data structure

**Data extracted includes:**

* **hostname:** Device hostname
* **system\_ip:** System IP address
* **device\_id:** Unique device identifier (UUID)
* **site\_id:** Site identification number
* **device\_model:** Hardware model information
* **device\_type:** Device type (vedge, vsmart, vbond, vmanage)
* **reachability:** Current reachability status
* **status:** Operational status
* **version:** Software version
* **certificate\_validity:** Certificate validation status
* **uptime:** Device uptime information
* **control\_connections:** Number of control connections
* **bfd\_sessions\_up:** Active BFD sessions count
* **bfd\_sessions\_down:** Inactive BFD sessions count

#### **Task 9: Complete Data Export (JSON)**

**Purpose:** Saves full device data in JSON format

**Generated file:** **all\_devices\_complete.json**

**What it does:**

* Exports complete device data with all fields
* Formats JSON for readability (pretty print)
* Preserves all vManage data fields
* Creates comprehensive backup of device information

#### **Task 10: Simplified Data Export (JSON)**

**Purpose:** Saves filtered device data in JSON format

**Generated file:** **all\_devices\_simplified.json**

**What it does:**

* Exports curated device fields only
* Provides cleaner data structure for analysis
* Formats JSON for human readability
* Focuses on key operational parameters

#### **Task 11: CSV Report Generation**

**Purpose:** Creates spreadsheet-compatible device report

**Generated file:** **all\_devices.csv**

**What it does:**

* Exports device data in CSV format
* Includes header row with field names
* Formats data for Excel/spreadsheet import
* Enables easy data filtering and sorting

#### **Task 12: Device Count Display**

**Purpose:** Shows total number of devices found

**What it displays:**

* Total count of devices in inventory
* Quick verification of retrieval success
* Immediate feedback on inventory size

#### **Task 13: Device Type Summary**

**Purpose:** Groups devices by type for overview

**What it displays:**

* Device count per type (vedge, vsmart, vbond, vmanage)
* Distribution of device types in network
* Quick inventory categorization
* Handles missing device types gracefully

#### **Task 14: Reachability Summary**

**Purpose:** Groups devices by reachability status

**What it displays:**

* Device count per reachability status
* Network connectivity overview
* Quick identification of unreachable devices
* Status distribution for monitoring

## **Output Files**

The playbook generates three output files:

* **all\_devices\_complete.json:** Full device data from vManage API
* **all\_devices\_simplified.json:** Key device fields in clean format
* **all\_devices.csv:** Spreadsheet-compatible device inventory