# **SD-WAN Device Details Playbook Documentation**

## **Overview**

The **get\_device\_details.yml** playbook is an Ansible automation script designed to retrieve detailed information about devices in Cisco SD-WAN environments. This playbook connects to the vManage controller and extracts comprehensive device data for analysis, monitoring, and documentation purposes.

## **Use Case**

**Use Case: Get device details - Retrieve specific device information**

This playbook addresses the need to:

* Retrieve complete inventory of all devices in the SD-WAN environment
* Extract detailed configuration and status information for specific devices
* Monitor device health and operational status
* Generate device reports for documentation and analysis
* Provide automated device information collection for troubleshooting

## **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"

device\_uuid: "{{ device\_id | default('') }}"

### **Directory Structure**

The playbook creates the following directory structure:

playbook\_directory/

├── get\_device\_details.yml

└── generated/

├── all\_devices.json

├── device\_details\_[uuid].json

├── device\_status\_[uuid].json

└── device\_hardware\_[uuid].json

## **Detailed 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 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 collection
* 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
* Uses form-based authentication with provided credentials
* Handles both **200** and **302** status codes for authentication
* Stores authentication cookies for subsequent API calls
* Sets **60-second timeout** to handle slow connections

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

**Purpose:** Obtains CSRF protection token required for API calls

**What it does:**

* Makes a GET request to **/dataservice/client/token** endpoint
* Uses authentication cookies from previous step
* Retrieves security token for API request headers
* Stores token for use in subsequent data requests

#### **Task 5: Device Inventory Collection (Conditional)**

**Purpose:** Retrieves complete list of all network devices

**API endpoint:** /dataservice/device

**Generated content:**

* Complete device inventory with basic information
* Device system IPs and hostnames
* Device types and operational status
* Device UUIDs for detailed queries

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

**Execution condition:** When no specific device UUID is provided

#### **Task 6: Device Inventory Storage**

**Purpose:** Saves complete device list to standardized location

**What it does:**

* Converts JSON response to formatted output
* Saves to **all\_devices.json** in generated directory
* Preserves all device metadata and identifiers
* Creates baseline inventory for reference

#### **Task 7: Device List Display**

**Purpose:** Shows available devices for user selection

**Generated output:**

* Device system IPs and hostnames
* Device types and classifications
* Device UUIDs for detailed queries
* Formatted display for easy device identification

#### **Task 8: Detailed Device Information (Conditional)**

**Purpose:** Retrieves comprehensive device configuration and status

**API endpoint:** /dataservice/device/details?deviceId=[uuid]

**Generated content:**

* Complete device configuration details
* Interface configurations and status
* Policy assignments and configurations
* Connectivity and routing information

**Generated file:** **device\_details\_[uuid].json**

**Execution condition:** When specific device UUID is provided

#### **Task 9: Device System Status Collection**

**Purpose:** Retrieves current operational status and performance metrics

**API endpoint:** /dataservice/device/system/status?deviceId=[uuid]

**Generated content:**

* CPU and memory utilization
* Service status and health indicators
* System uptime and performance metrics
* Operational state information

**Generated file:** **device\_status\_[uuid].json**

#### **Task 10: Device Hardware Information**

**Purpose:** Collects hardware specifications and health status

**API endpoint:** /dataservice/device/hardware/status?deviceId=[uuid]

**Generated content:**

* Physical component specifications
* Temperature and environmental readings
* Power supply status and consumption
* Hardware health and diagnostic information

**Generated file:** **device\_hardware\_[uuid].json**

#### **Task 11: Completion Notification**

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

**What it displays:**

* Success confirmation message
* Complete file paths for generated reports
* Summary of collected information types
* Location reference for user access

## **Execution Methods**

### **Complete Device Inventory**

To retrieve all devices in the network:

ansible-playbook get\_device\_details.yml

### **Specific Device Details**

To retrieve detailed information for a specific device:

ansible-playbook get\_device\_details.yml -e device\_id=[device\_uuid]

## **Generated Report Contents**

### **All Devices Report (all\_devices.json)**

The device inventory report includes:

* **System IP Addresses:** Management IP addresses for each device
* **Hostnames:** Device names and identifiers
* **Device Types:** Router, switch, or other device classifications
* **UUIDs:** Unique device identifiers for detailed queries
* **Operational Status:** Current device state and connectivity

### **Device Details Report (device\_details\_[uuid].json)**

The detailed device report includes:

* **Configuration Settings:** Complete device configuration
* **Interface Details:** Physical and logical interface information
* **Policy Assignments:** Applied security and routing policies
* **Template Information:** Device and feature template assignments
* **Connectivity Data:** Control and data plane connection status

### **Device Status Report (device\_status\_[uuid].json)**

The device status report includes:

* **Performance Metrics:** CPU, memory, and bandwidth utilization
* **Service Status:** Running services and process information
* **Health Indicators:** System health and diagnostic data
* **Connection Status:** Control connection and tunnel states

### **Device Hardware Report (device\_hardware\_[uuid].json)**

The hardware report includes:

* **Component Information:** Physical hardware specifications
* **Environmental Data:** Temperature, power, and fan status
* **Health Status:** Hardware component operational state
* **Diagnostic Information:** Hardware-level diagnostic results

## **Variable File Configuration**

**Note:** There is no separate variable file required for this playbook. All configuration is handled through environment variables set at the execution level, making the playbook portable and secure.

**Required Fields**

None - All required values are obtained from environment variables

**Optional Fields**

* **device\_id:** Specific device UUID for detailed information retrieval (provided via extra vars)