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

## **Overview**

The **device\_counter.yml** playbook is an Ansible automation script designed to collect device statistics counters from Cisco SD-WAN environments. This playbook connects to the vManage controller to get device information and retrieve performance counters from all devices in the network.

## **Use Case**

**Use Case 13: Get device counters - Retrieve device statistics counters**

This playbook addresses the need to:

* Get performance statistics from all SD-WAN devices
* Monitor device health and performance metrics
* Collect counter data for troubleshooting network issues
* Export device statistics for analysis and reporting
* Create automated monitoring for network performance tracking

## **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 | (required) |

## **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/

├── device\_counter.yml

└── generated/

├── device\_counters.json

├── devices\_list.json

└── device\_counters\_summary.csv

## **Task Analysis**

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

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

**What it does:**

* Checks that **VMANAGE\_HOST**, **VMANAGE\_USERNAME**, and **VMANAGE\_PASSWORD** are set
* Stops the playbook immediately if any required environment variable is missing
* Prevents failed connections due to missing credentials
* Shows clear error messages for troubleshooting

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

**Purpose:** Creates the output directory for generated files

**What it does:**

* Creates the **generated** directory relative to the playbook location
* Sets proper permissions (755) for file access
* Makes sure the output location exists before saving files
* Creates parent directories if they don't exist

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

**Purpose:** Checks if the vManage controller can be reached before trying to get data

**What it does:**

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

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

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

**What it does:**

* Checks if the connection test returned **HTTP 200** status
* Stops the playbook with error message if vManage cannot be reached
* Prevents trying to get data when connection problems exist
* Provides clear error messages for fixing issues

#### **Task 5: Get All Devices List**

**Purpose:** Gets information about all devices in the SD-WAN network

**API endpoint used:**

/dataservice/device

**What it does:**

* Connects to vManage using provided credentials
* Gets a complete list of all devices in the network including:
  + Device IDs and system IP addresses
  + Device hostnames and types
  + Site IDs and locations
  + Connection status and reachability
  + Device versions and models
* Stores device information for processing counters

#### **Task 6: Get Device Counters for Each Device**

**Purpose:** Collects performance statistics from each device

**API endpoint used:**

/dataservice/device/counters?deviceId=[device\_id]

**What it does:**

* Goes through each device found in the previous step
* Makes API calls to get counter data from each device
* Handles devices that might be unreachable or offline
* Collects performance metrics and statistics
* Stores counter data for each device

#### **Task 7: Save Device Counters to JSON File**

**Purpose:** Creates detailed JSON report with all counter data

**Generated file:** **device\_counters.json**

**What it does:**

* Creates a structured JSON file with collection information
* Includes metadata like collection date and total devices
* Stores detailed counter data for each device
* Shows which devices were successfully queried
* Uses proper JSON formatting for easy reading

#### **Task 8: Save Device List to JSON File**

**Purpose:** Creates backup copy of complete device inventory

**Generated file:** **devices\_list.json**

**What it does:**

* Saves the complete device inventory from vManage
* Keeps original device information as received from API
* Provides reference for device details and status
* Uses JSON format for easy processing

#### **Task 9: Create Device Counters Summary CSV**

**Purpose:** Creates simple summary table for quick analysis

**Generated file:** **device\_counters\_summary.csv**

**What it does:**

* Makes a CSV table with key device information
* Shows device ID, system IP, hostname, type, and site ID
* Indicates device reachability status
* Shows success or failure for counter collection
* Counts how many counter records were collected for each device

## **Generated Files**

The playbook creates three output files:

### **1. device\_counters.json**

* Complete counter data in JSON format
* Includes collection metadata and device details
* Contains all performance statistics from each device
* Structured for programmatic analysis

### **2. devices\_list.json**

* Complete device inventory from vManage
* All device attributes and status information
* Original API response data
* Reference for device specifications

### **3. device\_counters\_summary.csv**

* Simple table format for quick review
* Shows basic device information and collection status
* Easy to open in Excel or other spreadsheet programs
* Good for quick health checks and status reporting

## **Usage Examples**

### **Manual Execution**

cd /path/to/playbook/directory

ansible-playbook device\_counter.yml

### **Setting Environment Variables**

export VMANAGE\_HOST=your-vmanage-server.com

export VMANAGE\_USERNAME=your-username

export VMANAGE\_PASSWORD=your-password

ansible-playbook device\_counter.yml

## **Expected Results**

After successful execution, you will have:

* Statistics counters from all reachable devices
* Device inventory with current status
* Summary report showing collection results
* Files saved in the **generated** directory for analysis