# **SD-WAN API Ready Playbook Documentation**

## **Overview**

The SD-WAN API Ready playbook is an Ansible automation script designed to check the readiness of Cisco SD-WAN vManage API endpoints. This playbook validates that critical API services are accessible and responding correctly on vManage version 20.15.

## **File Name**

api\_ready.yml

## **Use Case**

**Use Case 8: api\_ready - Check API readiness**

## **Purpose**

* Verify vManage API endpoints are accessible and functional
* Validate authentication mechanisms are working
* Check core SD-WAN services availability
* Provide simple pass/fail status for API readiness

## **Detailed Task Analysis**

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

* - name: Validate environment variables are set

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

**What it does:**

* Checks that vmanage\_host, vmanage\_username, vmanage\_password, and vmanage\_port are set
* Fails the playbook immediately if any critical environment variables are missing
* Prevents failed API checks due to missing credentials

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

* - name: Create generated directory

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

**Generated folder:**

* {{ playbook\_dir }}/../generated - Simple output directory for results

### **Task 3: Authentication Endpoint Check**

* - name: Check API authentication endpoint

**Purpose:** Validates the authentication mechanism is working

**What it does:**

* Tests POST request to /j\_security\_check endpoint
* Uses form-based authentication with username/password
* Accepts status codes 200 or 302 as success
* 30-second timeout for response

### **Task 4: System Status Endpoint Check**

* - name: Check system status endpoint

**Purpose:** Verifies system controller information is accessible

**What it does:**

* Tests GET request to /dataservice/system/device/controllers
* Uses basic authentication
* Validates system services are responding
* Returns controller status information

### **Task 5: Device Inventory Endpoint Check**

* - name: Check device inventory endpoint

**Purpose:** Confirms device management APIs are functional

**What it does:**

* Tests GET request to /dataservice/device
* Validates device inventory services
* Ensures device management functionality is available

### **Task 6: Template Endpoint Check**

* - name: Check template endpoint

**Purpose:** Verifies template management APIs are accessible

**What it does:**

* Tests GET request to /dataservice/template/device
* Validates template services are functional
* Ensures configuration template access is working

### **Task 7: Policy Endpoint Check**

* - name: Check policy endpoint

**Purpose:** Confirms policy management APIs are operational

**What it does:**

* Tests GET request to /dataservice/template/policy/vedge
* Validates policy services are available
* Ensures policy management functionality is accessible

### **Task 8: Overall API Readiness Determination**

* - name: Determine overall API readiness

**Purpose:** Calculates overall API readiness status

**Generated result:**

* Sets api\_ready fact to true only if ALL endpoints pass
* Requires all 5 endpoint checks to return successful status codes
* Provides single boolean result for API readiness

### **Task 9: Results File Creation**

* - name: Save API readiness results

**Purpose:** Creates detailed results file for reference

**Generated file:** api\_ready\_results.txt in generated directory

**File contents:**

* Overall API readiness status (TRUE/FALSE)
* Individual endpoint test results (PASS/FAIL)
* HTTP status codes for each endpoint
* vManage host information
* Check completion timestamp

### **Task 10: Status Display**

* - name: Display API readiness status

**Purpose:** Shows completion confirmation and results location

**Generated output:**

* Final API readiness status
* File location for detailed results

## **API Endpoints Tested**

| **Endpoint** | **Purpose** | **Expected Status** | **Method** |
| --- | --- | --- | --- |
| /j\_security\_check | Authentication | 200/302 | POST |
| /dataservice/system/device/controllers | System Status | 200 | GET |
| /dataservice/device | Device Inventory | 200 | GET |
| /dataservice/template/device | Templates | 200 | GET |
| /dataservice/template/policy/vedge | Policies | 200 | GET |

## **Success Criteria**

**API is considered READY when:**

* Authentication endpoint responds with 200 or 302
* All other endpoints respond with 200
* No timeout or connection errors occur
* All 5 endpoint checks pass

**API is considered NOT READY when:**

* Any endpoint fails to respond
* Authentication fails
* Timeout errors occur
* Any endpoint returns error status codes

## **Output Files**

### **Generated Directory Structure**

* generated/
* └── api\_ready\_results.txt

### **Sample Output File Content**

* API Readiness Check Results
* ==========================
* vManage Host: vmanage-amfament-prod.sdwan.cisco.com
* Overall API Ready: TRUE
* Endpoint Test Results:
* - Authentication: PASS (Status: 302)
* - System Status: PASS (Status: 200)
* - Device Inventory: PASS (Status: 200)
* - Templates: PASS (Status: 200)
* - Policies: PASS (Status: 200)
* Check completed: 2025-08-19T10:30:45Z

## **Environment Variables Required**

| **Variable** | **Description** | **Default Value** |
| --- | --- | --- |
| VMANAGE\_HOST | vManage hostname or IP | vmanage-amfament-prod.sdwan.cisco.com |
| VMANAGE\_USERNAME | vManage username | automation |
| VMANAGE\_PASSWORD | vManage password | (required) |
| VMANAGE\_PORT | vManage HTTPS port | 443 |

## **Pipeline Integration**

### **Manual Execution Steps**

1. Navigate to your GitLab project
2. Go to **Code > Pipelines**
3. Click **Run Pipeline**
4. Select api\_ready.yml from the dropdown
5. Click **Run Pipeline** to trigger execution

### **Execution Results**

* Creates simple output file in generated/ directory
* No complex reporting or multiple directories
* Clean, straightforward results for API status verification

## **Error Handling**

The playbook includes comprehensive error handling:

* All API calls use ignore\_errors: true
* Timeout protection (30 seconds per endpoint)
* Graceful handling of authentication failures
* Status code validation for each endpoint
* Overall readiness calculation based on all checks

## **Use Cases**

This playbook is useful for:

* Pre-automation checks before running other SD-WAN tasks
* Health monitoring of vManage API services
* Troubleshooting API connectivity issues
* Validating system readiness after maintenance
* Integration testing in CI/CD pipelines