# **SD-WAN API Readiness Check Playbook Documentation**

## **Overview**

The **api\_ready.yml** playbook is an Ansible automation script designed to verify the readiness and availability of Cisco SD-WAN vManage API endpoints. This playbook performs comprehensive health checks on multiple API endpoints to ensure the vManage controller is fully operational and ready to accept API requests before executing other automation tasks.

## **Use Case**

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

This playbook addresses the need to:

* Verify vManage API availability before running automation tasks
* Check multiple API endpoints for comprehensive readiness assessment
* Validate authentication and connectivity to the vManage controller
* Generate readiness reports for troubleshooting and monitoring
* Ensure API stability before performing configuration changes

## **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:

# Python interpreter fix

ansible\_python\_interpreter: /usr/bin/python3

# Use environment variables for credentials

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"

# Output directory

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

### **Directory Structure**

The playbook creates the following directory structure:

playbook\_directory/

├── api\_ready.yml

└── generated/

├── api\_readiness\_check.txt

└── api\_readiness\_results.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 execution failures due to missing credentials
* Provides clear error messages for troubleshooting

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

**Purpose:** Creates the output directory for readiness 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 report generation
* Creates parent directories if they don't exist

#### **Task 3: Check vManage API Server Status**

**Purpose:** Verifies the API server status endpoint

**What it does:**

* Makes a REST API call to **/dataservice/server/status**
* Uses basic authentication with provided credentials
* Sets **30-second timeout** for response
* Ignores SSL certificate validation for internal certificates
* Stores status results for compilation

#### **Task 4: Check vManage System Readiness**

**Purpose:** Validates system controller endpoint availability

**What it does:**

* Makes a REST API call to **/dataservice/system/device/controllers**
* Verifies system-level API readiness
* Checks controller discovery and communication
* Essential for confirming core API functionality

#### **Task 5: Check vManage Device Dashboard Status**

**Purpose:** Verifies device dashboard API endpoint

**What it does:**

* Makes a REST API call to **/dataservice/device/dashboard/status**
* Validates device monitoring capabilities
* Checks dashboard data availability
* Confirms device statistics API access

#### **Task 6: Check vManage Alarms Endpoint**

**Purpose:** Validates alarm monitoring API availability

**What it does:**

* Makes a REST API call to **/dataservice/alarms**
* Verifies alarm retrieval functionality
* Checks monitoring system readiness
* Confirms event tracking capabilities

#### **Task 7: Check vManage Template Endpoint**

**Purpose:** Verifies template management API access

**What it does:**

* Makes a REST API call to **/dataservice/template/feature**
* Validates template retrieval capabilities
* Checks configuration template availability
* Essential for template-based operations

#### **Task 8: Check vManage Policy Endpoint**

**Purpose:** Validates policy management API availability

**What it does:**

* Makes a REST API call to **/dataservice/template/policy/list**
* Verifies policy retrieval functionality
* Checks policy configuration access
* Confirms policy management readiness

#### **Task 9: Check vManage vEdges Endpoint**

**Purpose:** Verifies edge device API endpoint

**What it does:**

* Makes a REST API call to **/dataservice/system/device/vedges**
* Validates vEdge device access
* Checks device inventory availability
* Confirms edge management capabilities

#### **Task 10: Compile API Readiness Results**

**Purpose:** Aggregates all endpoint check results

**What it does:**

* Compiles status from all API endpoint checks
* Determines overall API readiness status
* Creates structured result data
* Calculates endpoint-specific HTTP status codes

#### **Task 11: Save API Readiness Results to File**

**Purpose:** Creates human-readable readiness report

**Generated file:** **api\_readiness\_check.txt**

**What it does:**

* Creates formatted text report with all endpoint statuses
* Includes HTTP status codes for each endpoint
* Provides timestamp and connection details
* Saves to **generated/api\_readiness\_check.txt**

#### **Task 12: Save API Results as JSON**

**Purpose:** Creates machine-readable results file

**Generated file:** **api\_readiness\_results.json**

**What it does:**

* Exports results in JSON format for automation
* Includes all endpoint statuses and codes
* Provides structured data for parsing
* Saves to **generated/api\_readiness\_results.json**

#### **Task 13: Display API Readiness Status**

**Purpose:** Provides execution summary

**What it displays:**

* Overall API readiness status (Ready/Not Ready)
* Location of saved results files
* Quick visual confirmation of completion

## **Report Contents**

The generated reports include:

* **Server Status:** API server availability and health
* **System Ready:** Core system API functionality
* **Dashboard Status:** Device monitoring capabilities
* **Alarms Endpoint:** Event and alarm monitoring
* **Templates Endpoint:** Configuration template access
* **Policies Endpoint:** Policy management availability
* **vEdges Endpoint:** Edge device management status
* **Overall Status:** Aggregated API readiness assessment

## **Expected Results**

### **Successful API Ready Status**

All endpoints return HTTP 200 status codes, indicating full API availability.

### **Partial API Ready Status**

Some endpoints may return 403 (Forbidden) or other error codes, particularly in sandbox environments with restricted permissions.

### **API Not Ready Status**

Critical endpoints fail to respond or return error codes, indicating the API is not fully operational.