UNLOCKING EFFICIENCY WITH CONTROL-M AUTOMATION API



Introduction

In the rapidly evolving digital world, businesses are always looking for ways to optimize processes, minimize manual tasks, and boost overall efficiency. For those that depend on job scheduling and workload automation, *Control-M* from BMC Software has been a reliable tool for years. Now, with the arrival of the *Control-M Automation API*, organizations can elevate their automation strategies even further. In this blog post, we'll delve into what the Control-M Automation API offers, the advantages it brings, and how it can help revolutionize IT operations.

What is the Control-M Automation API?

The Control-M Automation API from BMC Software demonstrates how developers can use Control-M to automate workload scheduling and management. Built on a RESTful architecture, the API enables an API-first, decentralized method for building, testing, and deploying jobs and workflows. It offers services for managing job definitions, deploying packages, provisioning agents, and setting up host groups, facilitating seamless integration with various tools and workflows.

With the API, you can:

- Streamline job submission, tracking, and control through automation.

- Connect Control-M seamlessly with DevOps tools such as Jenkins, GitLab, and Ansible.
- Develop customized workflows and applications to meet specific business requirements.
- Access real-time insights and analytics to support informed decision-making.

Key Benefits of Using Control-M Automation API

- **Infrastructure-as-Code (IaC) for Workload Automation**
  Enables users to define jobs as **code** using **JSON**, allowing for version control and better collaboration. It also supports automation through **GitOps workflows**, making workload automation an integral part of CI/CD pipelines.

- **RESTful API for Programmatic Job Management**
  Provides a RESTful API to create, update, delete, and monitor jobs from any programming language **(Python, Java, PowerShell, etc.)**. It allows teams to automate workflows without relying on a graphical interface, enabling **CI/CD** integration and process automation.

- **Enhanced Automation Capabilities**
  By leveraging the Control-M Automation API, organizations can automate routine tasks, decreasing reliance on manual processes and mitigating the potential for human error. This capability is particularly valuable for managing intricate, high-volume workflows.

- **Seamless Integration**
  By serving as a bridge between Control-M and external tools, the Control-M Automation API enables effortless integration with CI/CD pipelines, cloud services, and third-party applications—streamlining workflows into a unified automation environment.

- **Improved Agility**
  Through Control-M Automation API integration, organizations gain the flexibility to accelerate application deployments and dynamically scale operations, ensuring responsive adaptation to market changes.

- **Real-Time Monitoring and Reporting**
  The Control-M Automation API provides real-time access to job statuses, logs, and performance metrics. This enables proactive monitoring and troubleshooting, ensuring smoother operations.

- **Customization and Extensibility**
  The API provides the building blocks to develop purpose-built solutions matching your exact specifications, including custom visualization interfaces and integration with specialized third-party applications.

Use Cases for Control-M Automation API

- **DevOps Integration**
  Integrate Control-M with your DevOps pipeline to automate the deployment of applications and infrastructure. For example, you can trigger jobs in Control-M from Jenkins or GitLab, ensuring a seamless flow from development to production.

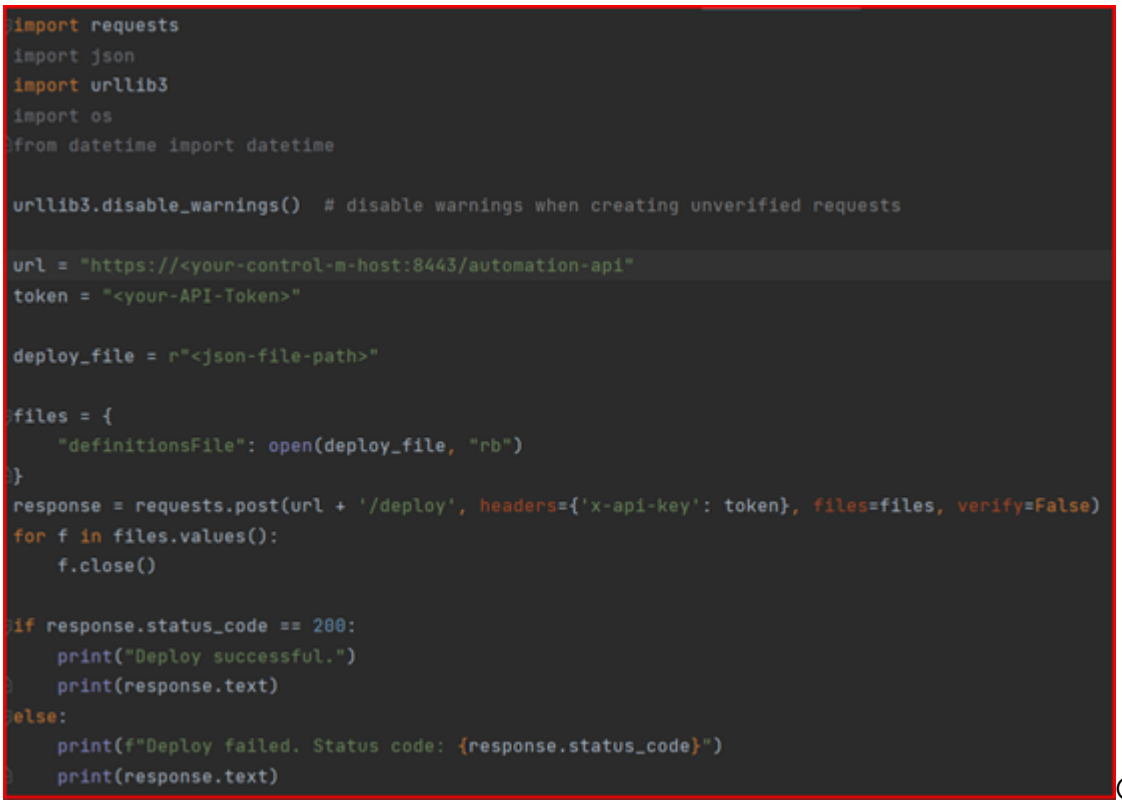
- **Cloud Automation**

Leverage the Control-M API to handle workloads across hybrid and multi-cloud setups. Streamline resource provisioning through automation, track cloud-based tasks, and maintain adherence to organizational policies.

- **Data Pipeline Automation**
  Automate data ingestion, transformation, and loading processes. The API can be used to trigger ETL jobs, monitor their progress, and ensure data is delivered on time.
- **Custom Reporting and Analytics**
  Extract job data and generate custom reports for stakeholders. The API can be used to build dashboards that provide insights into job performance, SLA adherence, and resource utilization.
- **Event-Driven Automation**
  Set up event-driven workflows where jobs are triggered based on specific conditions or events. For example, you can automate the restart of failed jobs or trigger notifications when a job exceeds its runtime.

Example 1: Scheduling a Job with Python

Here is another example of using the Control-M Automation API to define and deploy a job in Control-M. For this, we'll use a Python script (you'll need a Control-M environment with API access set up).



```python
import requests
import json
import urllib3
import os
from datetime import datetime

urllib3.disable_warnings()  # disable warnings when creating unverified requests

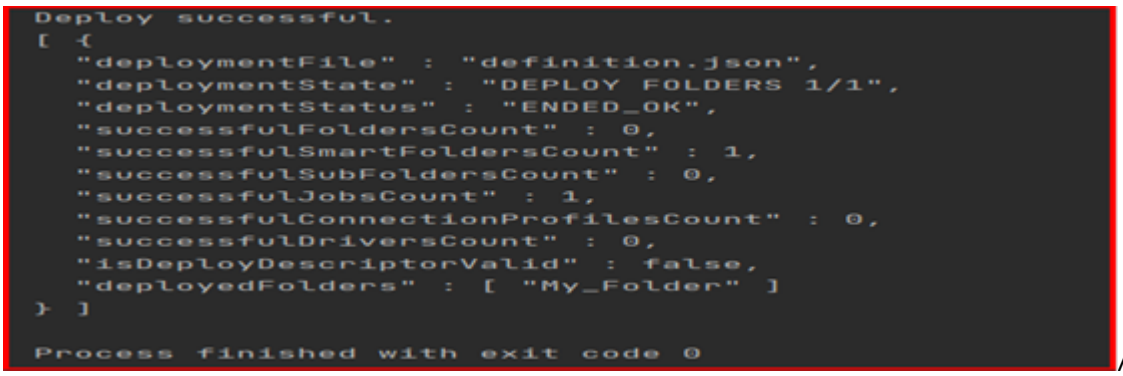
url = "https://<your-control-m-host:8443/automation-api"
token = "<your-API-Token>"

deploy_file = r"<json-file-path>"

files = {
    "definitionsFile": open(deploy_file, "rb")
}
response = requests.post(url + '/deploy', headers={'x-api-key': token}, files=files, verify=False)
for f in files.values():
    f.close()

if response.status_code == 200:
    print("Deploy successful.")
    print(response.text)
else:
    print(f"Deploy failed. Status code: {response.status_code}")
    print(response.text)
```

Output of the code shows the successful deployment of jobs/folder in Control-M.



And the folder is successfully deployed in Control-M which can be checked in GUI.



Example 2: Automating Job Submission with Python

Here's a simple example of how you can use the Control-M Automation API to submit a job using Python:

```python
import requests
import sys
import json

# Disable SSL warnings
import urllib3

urllib3.disable_warnings()

# === Configuration ===
url = "https://<your-control-m-host>:8443/automation-api"
token = "<your-API-token>"

# === Step 1: Get CLI arguments ===
if len(sys.argv) < 5:
    print("Usage: script.py <ctm> <folderName> <job_names> <order_date>")
    sys.exit(1)

ctm = sys.argv[1]
folderName = sys.argv[2]
job_names = sys.argv[3]
order_date = sys.argv[4]

# === Step 2: Order Job ===
order_payload = {
    "ctm": ctm,
    "folder": folderName,
    "jobs": job_names,
    "hold": "false",
    "ignoreCriteria": "true",
    "orderDate": order_date,
    "waitForOrderDate": "false",
    "orderIntoFolder": "New"
}
order_response = requests.post(
    f"{url}/run/order",
    headers={"x-api-key": token, "Accept": "application/json"},
    verify=False,
    json=order_payload
)

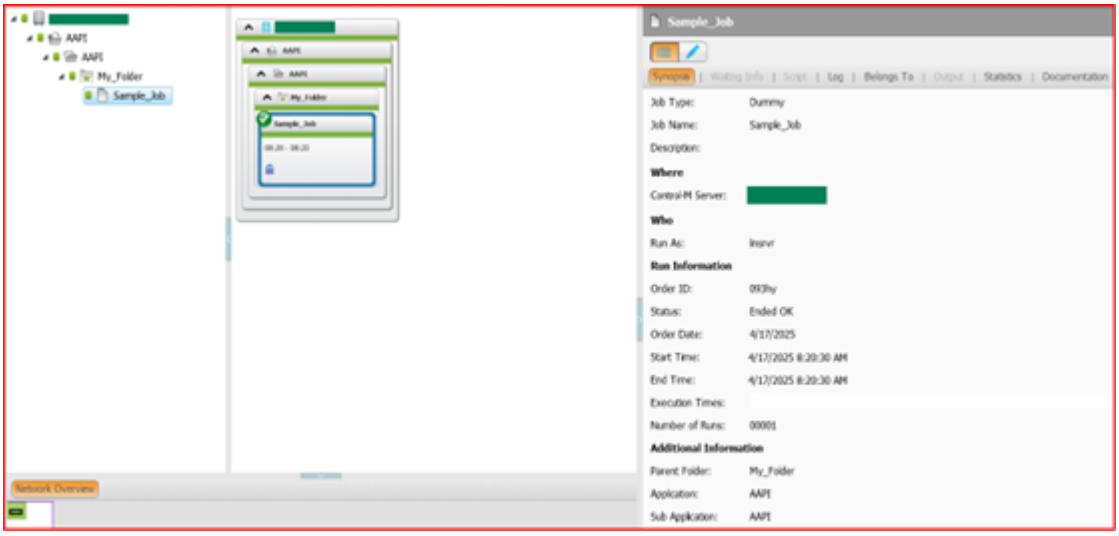
if order_response.status_code != 200:
    print(f"Order failed: {order_response.status_code}")
    print(order_response.text)
    sys.exit(1)

# === Step 3: Extract runId ===
run_id = order_response.json().get("runId")
if run_id:
    print(run_id)
else:
    print("runId not found in response")
```

Execute the above Python code and it will return the “RUN ID”.



The folder is successfully ordered and executed which can be checked in Control-M GUI.



Getting Started with Control-M Automation API

- **Access the API Documentation**
  BMC provides comprehensive documentation for the Control-M Automation API, including endpoints, parameters, and examples. Familiarize yourself with the documentation to understand the capabilities and limitations of the API.
- **Set Up Authentication**
  The API uses token-based authentication. Generate an API token from the Control-M GUI and use it to authenticate your requests.
- **Explore Sample Scripts**
  BMC offers sample scripts and code snippets in various programming languages (Python, PowerShell, etc.) to help you get started. Use these as a reference to build your own integrations.
- **Start with Simple Use Cases**
  Begin by automating simple tasks, such as job submission or status monitoring. Once you're comfortable, move on to more complex workflows.
- **Leverage Community and Support**
  Join the BMC community forums to connect with other users, share ideas, and troubleshoot issues. BMC also offers professional support services to assist with implementation.

Conclusion

The Control-M Automation API is a game-changer for organizations looking to enhance their automation capabilities. By enabling seamless integration, real-time monitoring, and custom workflows, the API empowers businesses to achieve greater efficiency and agility. Whether you're a developer, IT professional, or business leader, now is the time to explore the potential of the Control-M Automation API and unlock new levels of productivity.