

HyperSDK

API Reference & CI/CD Integration

150+ REST endpoints for VM discovery, export, migration, and management — with WebSocket streaming, webhook notifications, and native CI/CD pipeline integration.

REST + WebSocket — OpenAPI 3.0 — Python & TypeScript SDKs

API Overview

A comprehensive REST API for every HyperSDK capability, with real-time WebSocket support.

150+

REST Endpoints

JSON

Request/Response
Format

WS

WebSocket
Streaming

v1

Versioned API
(/api/v1/)



RESTful Design

Standard HTTP methods (GET, POST, PUT, DELETE) with consistent JSON request/response bodies. Proper HTTP status codes, pagination, and error responses with structured error objects.



WebSocket Streaming

Real-time job progress, log streaming, and event notifications over WebSocket connections. Subscribe to specific job IDs or receive all events for dashboard integration.



OpenAPI 3.0 Spec

Complete OpenAPI specification for all endpoints. Auto-generates client libraries, Swagger UI documentation, and Postman collections from the spec.



Authentication & Security

Bearer token and API key authentication. Rate limiting, request size limits, CORS headers, and security headers (CSP, HSTS) applied via middleware chain.

API Architecture

Client Request



Auth Middleware



Rate Limiter



Handler



Port Interface



Adapter

Core Endpoints

Essential endpoints for health monitoring, VM discovery, and job management.

System Endpoints

Method	Endpoint	Description
GET	/api/v1/health	Health check with component status
GET	/api/v1/status	Daemon status, uptime, version, and active jobs
GET	/api/v1/capabilities	Available providers, formats, and features
GET	/api/v1/metrics	Prometheus-format metrics export

Job Endpoints

Method	Endpoint	Description
POST	/api/v1/jobs	Submit a new export/migration job
GET	/api/v1/jobs	List all jobs with filtering and pagination
GET	/api/v1/jobs/{id}	Get job details, progress, and logs
POST	/api/v1/jobs/{id}/cancel	Cancel a running job gracefully
DELETE	/api/v1/jobs/{id}	Delete a completed job and its artifacts
GET	/api/v1/jobs/{id}/download	Download exported disk image

VM Discovery Endpoints

Method	Endpoint	Description
GET	/api/v1/vms	List VMs across all configured providers
GET	/api/v1/vms/{provider}	List VMs from a specific provider
GET	/api/v1/vms/{provider}/{id}	Get detailed VM info by ID

Advanced Endpoints

Schedules, webhooks, migrations, cost estimation, and carbon intensity APIs.

Schedule & Automation

Method	Endpoint	Description
POST	/api/v1/schedules	Create a scheduled job (cron or one-time)
GET	/api/v1/schedules	List all schedules with next-run times
DELETE	/api/v1/schedules/{id}	Remove a schedule
POST	/api/v1/webhooks	Register a webhook for job events
GET	/api/v1/webhooks	List registered webhooks
DELETE	/api/v1/webhooks/{id}	Remove a webhook

Migration & Cost

Method	Endpoint	Description
POST	/api/v1/migrations	Start an end-to-end migration workflow
GET	/api/v1/migrations/{id}	Get migration status across all phases
POST	/api/v1/cost/estimate	Estimate storage, egress, and compute costs
GET	/api/v1/carbon/intensity	Current carbon intensity for a region
GET	/api/v1/carbon/forecast	Forecast optimal low-carbon windows
POST	/api/v1/audit/query	Search audit logs with filters



Cost Estimation

Pre-migration cost calculator that factors in cloud storage, egress, compute hours, and format conversion time. Returns itemized breakdown per VM.



Carbon-Aware API

Real-time carbon intensity data and forecasts. Schedule migrations during low-carbon windows to reduce environmental impact without manual planning.

CI/CD Integration

Integrate HyperSDK into your existing CI/CD pipelines with simple HTTP calls.



Jenkins

Use the HTTP Request plugin or curl in a shell step. Poll job status until completion, then archive the exported artifact.



GitLab CI

Define migration stages in .gitlab-ci.yml. Use curl to submit jobs and wget to download artifacts. GitLab variables for credentials.



GitHub Actions

Custom action or curl-based workflow steps. Use GitHub Secrets for API tokens. Artifact upload for exported disk images.

Example: GitHub Actions Workflow

```
# .github/workflows/migrate-vm.yml
name: VM Migration
on: workflow_dispatch
jobs:
  migrate:
    runs-on: ubuntu-latest
    steps:
      - name: Submit export job
        run: |
          JOB_ID=$(curl -s -X POST \
            -H "Authorization: Bearer ${ secrets.HYPERSDK_TOKEN }" \
            -H "Content-Type: application/json" \
            -d '{"provider":"vsphere","vm":"web-server","format":"qcow2"}'
            ${ secrets.HYPERSDK_URL }/api/v1/jobs | jq -r '.id')
          echo "JOB_ID=$JOB_ID" >> $GITHUB_ENV

      - name: Wait for completion
        run: |
          while true; do
            STATUS=$(curl -s -H "Authorization: Bearer ${ secrets.HYPERSDK
              ${ secrets.HYPERSDK_URL }/api/v1/jobs/$JOB_ID | jq -r '.sta
```

```
    [ "$STATUS" = "completed" ] && break
    [ "$STATUS" = "failed" ] && exit 1
    sleep 30
done

- name: Download artifact
  run: |
    curl -o vm-export.qcow2 -H "Authorization: Bearer ${ secrets.HYP
      ${ secrets.HYPERSDK_URL }}/api/v1/jobs/$JOB_ID/download
```

Any tool that can make HTTP requests can integrate with HyperSDK. No special plugins, agents, or SDKs required — though we provide them for convenience.



SDK Clients

Official SDK clients and OpenAPI code generation for any language.



Python SDK

pip install hypersdk — Pythonic client with async support, type hints, and context managers. Wraps all REST endpoints with retry logic and progress callbacks.



TypeScript SDK

npm install @hypersdk/client — Fully typed TypeScript/JavaScript client. Promise-based API with WebSocket streaming support for real-time progress in web apps.



OpenAPI Codegen

Generate clients for Java, C#, Ruby, Rust, or any language using the OpenAPI 3.0 spec with openapi-generator, swagger-codegen, or oapi-codegen.

Python SDK Example

```
from hypersdk import HyperSDKClient

client = HyperSDKClient(url="https://hypersdk.example.com", token="...")

# Discover VMs
vms = client.vms.list(provider="vsphere", power_state="poweredOn")
for vm in vms:
    print(f"{vm.name}: {vm.cpus} vCPUs, {vm.memory_mb} MB")

# Submit export job
job = client.jobs.create(
    provider="vsphere",
    vm_name="web-server-01",
    output_format="qcow2",
)

# Wait for completion with progress
job.wait(callback=lambda p: print(f"{p.percent}% - {p.throughput_mbps} MB/s"))
```

```
# Download result
job.download("/exports/web-server-01.qcow2")
```

TypeScript SDK Example

```
import { HyperSDKClient } from '@hypersdk/client';

const client = new HyperSDKClient({
  url: 'https://hypersdk.example.com',
  token: process.env.HYPERSDK_TOKEN,
});

// List VMs with streaming progress
const vms = await client.vms.list({ provider: 'azure' });
const job = await client.jobs.create({
  provider: 'azure', vmName: 'db-server', format: 'qcow2',
});

// WebSocket progress streaming
job.onProgress((progress) => {
  console.log(`${progress.percent}% complete`);
});
await job.waitForCompletion();
```

Automate Everything

HyperSDK's API-first design means every operation available in the UI and CLI is also available via REST, WebSocket, and native SDK clients.

Build your own automation on top of a solid foundation.