DCDN Cloud — Technical Guide v6

DCDN Cloud — Technical Documentation v3

Last updated: 2026-03-23

Table of Contents

---

1. Architecture Overview

DCDN Cloud is a decentralized CDN and cloud platform. Unlike legacy CDN providers (single company), DCDN runs on independent node operators across multiple regions.

Stack: - **Coordinator:** FastAPI (Python) on dedicated Hetzner server (65.109.91.24) - **Database:** PostgreSQL - **Nodes:** 7 nodes across EU, US, South America — Rust-based CDN proxy - **DNS:** CoreDNS on ns1.dcdncloud.com + ns2.dcdncloud.com - **SSL:** Let’s Encrypt auto-provisioning - **Payments:** Stripe (USD) + RSP token (crypto)

How requests flow: 1. User’s browser → DNS lookup → DCDN nameservers 2. DCDN nameservers → nearest node IP 3. Node receives request → checks cache → if miss, fetches from origin 4. Response served with caching headers, WAF rules applied

---

2. Getting Started

Step 1: Create an account

Go to https://dcdncloud.com/dashboard and register with email + password, or use Google Sign-In.

Step 2: Add your domain

1. Click “Domains” in the sidebar
2. Click “+ Add Domain”
3. Enter your domain name and origin IP
4. DCDN creates DNS records automatically

Step 3: Change your nameservers

At your domain registrar, change nameservers to:

```
ns1.dcdncloud.com (65.109.91.24)
ns2.dcdncloud.com (187.77.61.68)
```

DNS propagation takes 5-60 minutes.

Step 4: Done!

Your site is now served through DCDN. SSL is provisioned automatically.

---

3. CDN & Domains

Adding a Domain

Dashboard: Domains → + Add Domain **API:**

```
curl -X POST https://dcdncloud.com/api/v1/cdn/domains \
  -H "Authorization: Bearer YOUR_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{"domain": "example.com", "origin_ip": "1.2.3.4"}'
```

DNS Records

DCDN manages DNS for your domain. You can add/edit records: - A, AAAA, CNAME, MX, TXT, SRV, NS, CAA - Toggle “Proxied” to route through CDN or direct to origin

SSL/TLS

- **Free:** Wildcard SSL for all domains
- **Modes:** Off, Flexible, Full, Full (Strict)
- Certificates are auto-provisioned via Let’s Encrypt

Caching

- **Cache levels:** Standard, Aggressive, Bypass
- **Browser TTL:** 1 hour to 1 year
- **Edge TTL:** 1 hour to 30 days
- **Dev Mode:** Bypass all caching for 3 hours (debugging)

Page Rules

Create URL-based rules for caching, redirects, forwarding, and security overrides. Example: `*.example.com/api/*` → Cache Bypass

---

4. Image Optimization CDN

Automatically resize, convert, and optimize images on-the-fly. No configuration needed — just use the URL.

Usage

```
https://dcdncloud.com/cdn-cgi/image/?url=SOURCE_URL&w=WIDTH&h=HEIGHT&f=FORMAT&q=QUALITY&fit=FIT
```

Parameters

Parameter	Description	Default	Example
`url`	Source image URL (required)	—	`https://example.com/photo.jpg`
`w`	Width in pixels (1-4096)	original	`800`
`h`	Height in pixels (1-4096)	original	`600`
`f`	Format: `auto`, `webp`, `avif`, `jpeg`, `png`	`auto`	`webp`
`q`	Quality 1-100	`80`	`75`
`fit`	Resize mode: `cover`, `contain`, `fill`, `inside`	`cover`	`contain`

Examples

```
# Resize to 800px wide, auto WebP/AVIF
/cdn-cgi/image/?url=https://example.com/big-photo.jpg&w=800

# Thumbnail 200x200, JPEG quality 60
/cdn-cgi/image/?url=https://example.com/photo.jpg&w=200&h=200&f=jpeg&q=60

# Force PNG output
/cdn-cgi/image/?url=https://example.com/photo.jpg&w=400&f=png
```

Auto Format

When `f=auto`, the CDN checks the browser's `Accept` header: - Supports AVIF → serves AVIF (smallest) - Supports WebP → serves WebP - Otherwise → serves JPEG

Caching

Optimized images are cached for 7 days. Cache headers: `Cache-Control: public, max-age=604800`

Security

- Only `http://` and `https://` URLs allowed
- Internal IPs blocked (127.0.0.1, 10.x, 172.16.x, 169.254.x)
- Max source size: 20MB

---

5. Edge Functions (Serverless)

Deploy serverless functions at the edge. Like Edge Workers, but no vendor lock-in.

Supported Runtimes

Runtime	Version	Language
Node.js	18	JavaScript
Python	3.11	Python
Deno	Latest	TypeScript

Creating a Function

Dashboard: Edge Functions → + New Function → pick runtime → write code → Deploy

API:

```
curl -X POST https://dcdncloud.com/api/v1/edge/functions \
  -H "Authorization: Bearer YOUR_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "name": "my-api",
    "runtime": "python",
    "code": "import json\ndef handler(request):\n    return {\"status\": 200, \"headers\": {\"Content-Type\":
        \"application/json\"}, \"body\": json.dumps({\"hello\": \"world\"})}"
  }'
```

Invoking a Function

Each function gets a public URL (no auth required, like CF Workers):

```
https://dcdncloud.com/api/v1/edge/invoke/YOUR_FUNCTION_ID
```

Supports all HTTP methods: GET, POST, PUT, DELETE, PATCH.

Python Example

```
import json
from datetime import datetime

def handler(request):
```

```
    name = request.get("query", "").split("name=")[-1] if "name=" in request.get("query", "") else "World"
    return {
        "status": 200,
        "headers": {"Content-Type": "application/json"},
        "body": json.dumps({
            "message": f"Hello, {name}!",
            "timestamp": datetime.utcnow().isoformat(),
            "method": request.get("method"),
        })
    }
```

Node.js Example

```
export default {
    async fetch(request) {
        const url = new URL(request.url);
        return new Response(JSON.stringify({
            message: "Hello from DCDN Edge!",
            path: url.pathname,
        }), {
            headers: { "Content-Type": "application/json" },
        });
    }
};
```

Limits

Plan	Max Functions
Free	3
Standard	10
Pro	25
Business	100

Security

- Python functions run in a sandboxed namespace (exec/eval/open/**import** blocked)
- Node.js/Deno run in Docker containers (64MB RAM, 0.5 CPU, no network, read-only filesystem)
- 10 second timeout per invocation

---

6. Compute (VPS) & One-Click Apps

VPS Plans

Plan	vCPU	RAM	Disk	Price
Nano	1	512 MB	10 GB	$3/mo
Micro	1	1 GB	20 GB	$5/mo
Small	1	2 GB	40 GB	$8/mo
Medium	2	4 GB	60 GB	$15/mo
Large	4	8 GB	100 GB	$30/mo
XL	8	16 GB	200 GB	$60/mo

One-Click App Templates

Instead of a blank OS, deploy a pre-configured stack:

Template	What's Included	Min Plan
WordPress	Nginx + MySQL + PHP 8.2 + WP-CLI + Certbot	Micro
Ghost	Node.js + Ghost CMS + SQLite + Nginx + Certbot	Small
Next.js	Node.js 18 + PM2 + Nginx reverse proxy + Certbot	Micro
Static Site	Nginx web server + Certbot	Nano
Node.js	Node.js 18 + PM2 + Nginx + Certbot	Micro
Python/FastAPI	Python 3.11 + FastAPI + Uvicorn + Nginx + Certbot	Micro

How to deploy: 1. Dashboard → Instances (VPS) → pick a template from the grid 2. Choose plan and region 3. Click “Create Instance” 4. Your app is live in ~2 minutes with auto SSL

---

7. Uptime Monitoring

Monitor your websites and services. Get alerted when they go down.

Creating a Monitor

Dashboard: Uptime Monitoring → + Add Monitor

Field	Description
Name	Display name (e.g. “My Website”)
URL	Full URL to check (e.g. https://example.com)
Type	HTTP, HTTPS, or TCP
Interval	How often to check (minimum 30 seconds)
Alert Email	Email for down/up notifications

How It Works

- Background scheduler checks all monitors automatically
- If a monitor goes DOWN → incident created, alert sent
- When it recovers → incident resolved automatically
- Uptime percentage calculated (24h rolling)

Multi-Region Checks

Monitors are checked from 4 regions: - EU Central (Germany) - EU West (France) - US East (Virginia) - South America (Brazil)

Public Status Page

Each monitor has a public status URL:

```
https://dcdncloud.com/api/v1/monitoring/status/MONITOR_ID
```

Share this with your users as a status page.

Limits

Plan	Max Monitors
Free	2
Standard	5
Pro	15
Business	50

---

8. Email Forwarding

Forward emails from your domain without running a mail server.

Setup

1. Dashboard → Email Forwarding → + Add Route
2. Pick your domain
3. Enter source address (or * for catch-all)
4. Enter destination email (e.g. you@gmail.com)
5. Add MX record to your domain: `MX 10 mail.dcdncloud.com`

Examples

Source	Destination	Type
`info@yourdomain.com`	`you@gmail.com`	Single address
`*@yourdomain.com`	`you@gmail.com`	Catch-all
`support@yourdomain.com`	`team@company.com`	Single address

DNS Check

The dashboard shows if your MX records are correctly configured.

Limits

Plan	Max Routes
Free	5
Standard	25
Pro	100
Business	500

---

9. DDoS Protection

Always-on DDoS protection for all domains. No configuration needed.

Dashboard

The DDoS Protection page shows: - **Live stats:** Requests/sec, bandwidth, blocked requests, cache hit rate, threat score - **24h traffic chart:** Visual breakdown of total vs blocked vs challenged traffic - **Attack types:** HTTP Flood, DNS Amplification, SYN Flood, Slowloris breakdown - **Mitigation methods:** Rate Limiting, JS Challenge, IP Blacklist, GeoIP Block, WAF Rules - **Top traffic sources:** Country breakdown - **Recent attacks:** Table with type, severity, peak RPS, source IPs, status

Under Attack Mode

If you’re actively being attacked: 1. Dashboard → DDoS Protection → “Enable Under Attack Mode” 2. All visitors get a JavaScript challenge before accessing your site 3. Disable when the attack stops

What’s Protected

- Volumetric attacks (bandwidth floods)
- Protocol attacks (SYN flood, etc.)
- Application layer attacks (HTTP flood, slowloris)
- DNS amplification

---

10. Geo A/B Testing

Route traffic to different origins based on visitor country. Pro plan and above.

How It Works

1. Create a test for your domain
2. Add rules: “Visitors from US,CA → origin A” + “Visitors from DE,FR → origin B”
3. DCDN routes traffic based on GeoIP
4. Track impressions and conversions per variant

Use Cases

- Serve localized content from regional servers
- A/B test landing pages by geography
- Route EU users to GDPR-compliant infrastructure
- Serve different pricing pages by market

API

```
curl -X POST https://dcdncloud.com/api/v1/geo-ab/tests \
  -H "Authorization: Bearer YOUR_TOKEN" \
  -d '{
    "domain_name": "example.com",
    "name": "EU vs US landing page",
    "default_origin": "1.2.3.4",
    "rules": [
      {"countries": "US,CA", "origin": "5.6.7.8", "variant_name": "US"},
      {"countries": "DE,FR,GB", "origin": "9.10.11.12", "variant_name": "EU"}
    ]
  }'
```

---

11. high availability

DCDN is decentralized by design. No single entity can take your site down.

Features

- **Multi-Node Redundancy:** Content served from multiple independent operators
- **DNS Fallback:** Secondary nameservers in different regions
- **No Central Kill Switch:** Unlike legacy CDN providers, no single company controls the network
- **Geographic Distribution:** Nodes in 5+ regions across 4 continents
- **Transparent Operations:** Open sanctioned countries list, no editorial content moderation
- **Origin Encryption:** Full E2E encryption to your origin server

Enable

Dashboard → Domain Settings → Enable “high availability”

Or via API:

```
curl -X POST https://dcdncloud.com/api/v1/content moderation/enable \
  -H "Authorization: Bearer YOUR_TOKEN" \
  -d '{"domain_name": "example.com", "multi_node_redundancy": true, "dns_fallback": true}'
```

Included in all plans. This is a core feature, not an add-on.

---

12. Affiliate Program

Earn 80% recurring commission on every payment from customers you refer.

How It Works

1. Dashboard → Affiliate Program → Join
2. Get your unique referral link: `https://dcdncloud.com/?ref=YOUR_CODE`
3. Share the link
4. When someone registers via your link AND pays → you earn 80% of their payment
5. Recurring — you earn on every monthly renewal
6. Request payout when balance reaches $100

Dashboard

- Referral link (copy button)
- Total referrals, paying referrals
- Total earned, available balance
- Commission history (amount, %, status)
- Payout history

Rules

- No self-referral (your own account doesn’t count)

- Minimum payout: $100
- Payouts via Stripe (USD)
- Commission on ALL plan payments (Standard, Pro, Business, VPS add-ons)

---

13. Migrating from legacy CDN providers

Automatic Migration

1. Go to https://dcdncloud.com/migrate
2. Enter your legacy CDN providers API token (read-only)
3. Select domains to migrate
4. Preview DNS records, SSL settings
5. Click “Import to DCDN Cloud”
6. Change nameservers at your registrar

What’s Imported

- All DNS records (A, AAAA, CNAME, MX, TXT, etc.)
- SSL mode (mapped to DCDN equivalent)
- Domain configuration

What You Need

A legacy CDN providers API token with: - Zone:Read permission - DNS:Read permission

Create one at: https://dash.legacy CDN providers.com/profile/api-tokens

---

14. Security

Infrastructure

- SSH hardening: key-only auth, MaxAuthTries 3, no password auth
- Fail2ban on all 7 servers
- Unattended security upgrades enabled
- UFW firewall on all servers
- PostgreSQL: localhost only, scram-sha-256

Application

- CORS: strict origin validation (only dcdncloud.com)
- Rate limiting: login (3/10min), register (3/10min), API (20 burst)
- CSRF: SameSite cookies + Origin validation
- XSS: Content-Security-Policy headers on all domains
- SQL Injection: SQLAlchemy ORM (parameterized queries)
- SSRF: Internal IP blocking on Image CDN
- Node API: nginx IP whitelist (only known node IPs)

Headers (all domains)

```
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Strict-Transport-Security: max-age=31536000; includeSubDomains
Referrer-Policy: strict-origin-when-cross-origin
Permissions-Policy: camera=(), microphone=(), geolocation=()
Content-Security-Policy: [domain-specific]
```

Sanctioned Countries

23 countries blocked at the CDN node level (hardcoded in Rust WAF + coordinator push): CU, IR, KP, SY, RU, BY, MM, VE, SD, SO, YE, AF, CF, CD, LY, LB, IQ, ZW, ML, NI, HT, SS, ER

---

15. API Reference

Base URL: `https://dcdncloud.com/api/v1`

All authenticated endpoints require: `Authorization: Bearer YOUR_TOKEN`

Auth

Method	Endpoint	Description
POST	/auth/register	Register (email, password, referral_code)
POST	/auth/login	Login → JWT token
GET	/auth/verify-email?token=X	Verify email address

Domains

POST | /cdn/domains | Add domain |
GET | /cdn/domains | List domains |
PATCH | /cdn/domains/{id} | Update domain settings |

Image CDN

GET | /cdn-cgi/image/?url=...&w=...&f=... | Optimize image (public) |

Edge Functions

GET | /edge/runtimes | List runtimes (public) |
POST | /edge/functions | Create function |
GET | /edge/functions | List functions |
GET | /edge/functions/{id} | Get function + code |
PATCH | /edge/functions/{id} | Update function |
DELETE | /edge/functions/{id} | Delete function |
ANY | /edge/invoke/{id} | Invoke function (public) |

Compute

GET | /compute/templates | List app templates (public) |
GET | /compute/plans | List VPS plans (public) |
POST | /compute/instances | Create instance |
GET | /compute/instances | List instances |

Monitoring

POST | /monitoring/monitors | Create monitor |
GET | /monitoring/monitors | List monitors |
POST | /monitoring/monitors/{id}/check | Manual check |
GET | /monitoring/status/{id} | Public status page |
GET | /monitoring/regions | List check regions (public) |

Email

POST | /email/routes | Create forwarding rule |
GET | /email/routes | List routes |
DELETE | /email/routes/{id} | Delete route |

DDoS

GET | /ddos/overview | Protection overview |
GET | /ddos/live | Live traffic stats |
GET | /ddos/attacks | Attack history |
POST | /ddos/under-attack | Toggle Under Attack Mode |

Geo A/B

POST | /geo-ab/tests | Create test |
GET | /geo-ab/tests | List tests |
POST | /geo-ab/tests/{id}/rules | Add rule |
PATCH | /geo-ab/tests/{id}/toggle | Enable/disable |

Affiliate

POST | /affiliate/join | Join program |
GET | /affiliate/dashboard | Stats + earnings |
POST | /affiliate/request-payout | Request payout |

Migration

POST | /migrate/cf/zones | List CF domains |
POST | /migrate/cf/preview | Preview import |
POST | /migrate/cf/import | Execute migration |
GET | /migrate/compare | CF comparison (public) |

content moderation

GET | /content moderation/features | Feature list (public) |
POST | /content moderation/enable | Enable for domain |
GET | /content moderation/status/{domain} | Check status |

---

16. Pricing

Plan	Price	Domains	Bandwidth	VPS	Edge Functions	Monitors	Email Routes
Free	$0/mo	3	50 GB	—	3	2	5
Standard	$5/mo	10	200 GB	5	10	5	25
Pro	$15/mo	Unlimited	1 TB	10	25	15	100
Business	$59/mo	Unlimited	5 TB	50	100	50	500
Enterprise	Custom	Unlimited	Unlimited	Unlimited	500	200	Unlimited

All plans include: Wildcard SSL, DDoS protection, WAF, Image CDN, high availability, Affiliate program (80% commission)

DCDN Cloud — Decentralized infrastructure for the open internet. https://dcdncloud.com

---

AI Agent Platform

DCDN Cloud includes a built-in AI Agent hosting platform. Deploy AI agents on decentralized nodes with persistent memory, skills, and streaming support.

What is an AI Agent?

An AI agent is an automated assistant that runs on DCDN's decentralized nodes. You define its behavior (system prompt), choose a model, and interact with it via API or the dashboard. Agents have persistent memory — they remember previous conversations.

Quick Start

1. Create an Agent (Dashboard)

1. Go to **dcdncloud.com/dashboard**
2. Click **AI Agents** in the sidebar
3. Click **+ New Agent**
4. Fill in:
    - **Name:** e.g., "Customer Support Bot"
    - **Model:** Choose from available models (see Model Tiers below)
    - **System Prompt:** Define what the agent does, e.g., "You are a customer support agent for an e-

commerce store. Be helpful, concise, and friendly."
    - **Skills:** Select any skills the agent needs
5. Click **Create Agent**
6. Send a message to test it

2. Create an Agent (API)

```
curl -X POST https://dcdncloud.com/api/v1/agents \
  -H "Authorization: Bearer YOUR_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "name": "Support Bot",
    "model": "anthropic/claude-3-haiku",
    "system_prompt": "You are a helpful customer support agent.",
    "skills": ["web_search", "http_request"],
    "max_tokens": 4096,
    "temperature": 0.7
  }'
```

Response:

```
{
  "id": "abc123-...",
  "name": "Support Bot",
  "model": "anthropic/claude-3-haiku",
  "status": "active"
}
```

3. Run an Agent

Synchronous (wait for full response):

```
curl -X POST https://dcdncloud.com/api/v1/agents/AGENT_ID/run \
  -H "Authorization: Bearer YOUR_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{"message": "How do I return a product?"}'
```

Response:

```
{
  "run_id": "run-xyz...",
  "response": "To return a product, go to your order history...",
  "model": "anthropic/claude-3-haiku",
  "tokens": {"input": 45, "output": 120, "total": 165, "effective": 165},
  "cost": {"tier": "basic", "multiplier": 1, "real_cost_usd": 0.000161},
  "latency_ms": 850
}
```

Streaming (Server-Sent Events):

```
curl -X POST https://dcdncloud.com/api/v1/agents/AGENT_ID/stream \
  -H "Authorization: Bearer YOUR_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{"message": "Explain quantum computing"}'
```

Returns SSE stream:

```
data: {"content": "Quantum"}
data: {"content": " computing"}
data: {"content": " is a type of"}
...
data: {"done": true, "full_response": "Quantum computing is a type of..."}
```

4. Bring Your Own Key (BYOK)

If you have your own OpenAI/Anthropic API key, pass it in the request:

```
curl -X POST https://dcdncloud.com/api/v1/agents/AGENT_ID/run \
  -H "Authorization: Bearer YOUR_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
```

```
    "message": "Hello",
    "api_key": "sk-your-openai-key-here"
  }'
```

With BYOK, there are no token limits — you pay your provider directly.

Model Tiers

Not all models cost the same. We use a multiplier system:

Tier	Models	Multiplier	What it means
Basic	Claude 3 Haiku, GPT-4o Mini, Gemini Flash, DeepSeek	**1x**	1,000 tokens used = 1,000 from your balance
Standard	Claude Sonnet 4, GPT-4o, Gemini Pro	**5x**	1,000 tokens used = 5,000 from your balance
Premium	GPT-4, Claude Opus	**20x**	1,000 tokens used = 20,000 from your balance

Example: Pro plan includes 1,000,000 tokens/month. - With Claude Haiku (1x): ~1,000,000 tokens of conversation - With GPT-4o (5x): ~200,000 tokens of conversation - With GPT-4 (20x): ~50,000 tokens of conversation

Tip: Use Basic tier models for most tasks — they’re fast, cheap, and surprisingly capable. Switch to Standard/Premium only when you need it.

Agent Memory

Agents automatically remember conversations. Each message (user + assistant) is stored and included in the next request as context.

View memory:

```
GET /api/v1/agents/AGENT_ID/memory?limit=50
```

Clear memory:

```
DELETE /api/v1/agents/AGENT_ID/memory
```

Skills

Skills are built-in capabilities you can give to your agents:

Skill	What it does
`web_search`	Search the web for current information
`http_request`	Make HTTP requests to external APIs
`send_email`	Send emails via DCDN’s mail system
`json_parse`	Parse and transform JSON data
`code_execute`	Execute Python/JS code in a sandboxed environment
`memory_query`	Query the agent’s long-term memory

Add skills when creating an agent:

```
{"skills": ["web_search", "http_request", "code_execute"]}
```

Plan Limits

Plan	Agents	Runs/day	Included Tokens	BYOK
Free	1	100	0	Required
Standard ($5)	5	1,000	0	Required
Pro ($15)	25	10,000	1,000,000	Optional
Business ($59)	Unlimited	100,000	5,000,000	Optional

What happens when tokens run out? - You get a 429 error: “Token limit exceeded” - Options: upgrade your plan, wait for next month (resets on the 1st), or use BYOK

Usage Tracking

Check your usage anytime:

```
GET /api/v1/agents/usage
```

Response:

```
{
  "plan": "pro",
  "included_tokens": 1000000,
  "used_tokens": 345000,
  "remaining_tokens": 655000,
  "used_pct": 34.5,
  "this_month": {
    "raw_tokens": 69000,
    "runs": 423,
    "estimated_cost_usd": 0.0172
  }
}
```

Execution Logs

Every agent run is logged:

```
GET /api/v1/agents/AGENT_ID/logs?limit=20
```

Returns: input, output (truncated), tokens used, latency, model, status, and any errors.

API Reference

Endpoint	Method	Auth	Description
`/api/v1/agents`	POST		Create agent
`/api/v1/agents`	GET		List agents
`/api/v1/agents/{id}`	GET		Get agent details
`/api/v1/agents/{id}`	PATCH		Update agent
`/api/v1/agents/{id}`	DELETE		Delete agent
`/api/v1/agents/{id}/run`	POST		Execute (sync)
`/api/v1/agents/{id}/stream`	POST		Execute (SSE stream)
`/api/v1/agents/{id}/memory`	GET		Get memory
`/api/v1/agents/{id}/memory`	DELETE		Clear memory
`/api/v1/agents/{id}/logs`	GET		Execution logs
`/api/v1/agents/models/available`	GET		List models + tiers
`/api/v1/agents/usage`	GET		Token usage
`/api/v1/agents/skills/available`	GET		List skills
`/api/v1/agents/platform/stats`	GET		Public stats

Use Cases

- **Customer support chatbot** — answer FAQs, create tickets
- **Content generator** — blog posts, social media, emails
- **Data processor** — parse webhooks, transform data, call APIs
- **Monitoring assistant** — check status, alert on issues
- **Code helper** — review code, generate snippets, debug
- **Research agent** — search the web, summarize findings

DCDN Credits System

Overview

DCDN Credits are a promotional/bonus currency for the platform. **1 Credit = $1 USD** (fixed rate). Credits can only be granted by platform admins or authorized resellers — they cannot be purchased.

When a service is paid with credits, DCDN tokens are minted on-chain with the same 70/25/5 split as regular payments, ensuring node operators are compensated regardless of payment method.

Credit Flow

```
Admin/Reseller grants credits → User balance increases
User purchases service → Credits deducted (FIFO from oldest grant)
                       → DCDN token minted on-chain
                       → 70% Node Reward Pool
                       → 25% RSP Connector (→ 99% RSP holders, 1% RSP treasury)
                       →  5% LP/Treasury
```

Credit Rules (Business Logic)

- **Credits never expire** by default (`credit_expires_in_days: null`)
- **Token minting on credit spend**: When credits are spent, `on_credit_payment()` mints DCDN tokens with the standard 70/25/5 split — node operators are always compensated
- **No affiliate commission**: Credit-funded purchases do NOT trigger `record_commission()` — only Stripe/token payments do
- **Reseller tier progression**: Only Stripe/token payments increment `reseller.total_paid_sales` — credit purchases do not count via `record_reseller_paid_sale()`
- **Race condition protection**: `SELECT FOR UPDATE` on credit_accounts and credit_grants during spend operations
- **Promo code rate limit**: 5 redemption attempts per user per minute
- **Reseller self-redemption blocked**: Resellers cannot redeem their own promo codes
- **FIFO spending**: Credits deducted from oldest grant first
- **Promo code input**: Sanitized to `[A-Za-z0-9_-]` only

API Reference — Credits

Method	Endpoint	Auth	Description
GET	/credits/balance	User	Get credit balance and stats
GET	/credits/history	User	Transaction history (limit, offset)
GET	/credits/grants	User	Active grants with remaining amounts
POST	/credits/grant	Admin	Grant credits to a user
POST	/credits/grant/bulk	Admin	Grant credits to multiple users (max 500)
POST	/credits/revoke	Admin	Revoke credits from a user
GET	/credits/admin/overview	Admin	System-wide credit stats
GET	/credits/admin/user/{id}	Admin	User's credit details

Grant Credits (Admin)

```
POST /credits/grant
Authorization: Bearer <admin_token>
Content-Type: application/json

{
  "user_id": "uuid",
  "amount": 100,
  "campaign": "launch_promo_2026",
  "description": "Launch campaign bonus",
  "expires_in_days": 90
}
```

Bulk Grant (Admin)

```
POST /credits/grant/bulk
Authorization: Bearer <admin_token>
```

```
{
  "user_ids": ["uuid1", "uuid2", "uuid3"],
  "amount": 50,
  "campaign": "early_adopters",
  "expires_in_days": 60
}
```

Promo Code System

Overview

Promo codes allow users to redeem DCDN Credits. Codes can be created by admins (for platform campaigns) or by resellers (within their tier limits).

API Reference — Promo Codes

Method	Endpoint	Auth	Description
`POST`	`/promo/create`	Admin	Create a promo code
`GET`	`/promo/list`	Admin	List all promo codes
`POST`	`/promo/{id}/toggle`	Admin	Enable/disable a code
`POST`	`/promo/redeem`	User	Redeem a promo code

Create Promo Code (Admin)

```
POST /promo/create
Authorization: Bearer <admin_token>

{
  "code": "LAUNCH50",
  "credit_amount": 50,
  "campaign": "launch_2026",
  "max_uses": 1000,
  "per_user_limit": 1,
  "expires_in_days": 30,
  "credit_expires_in_days": null
}
```

Redeem Promo Code (User)

```
POST /promo/redeem
Authorization: Bearer <user_token>

{
  "code": "LAUNCH50"
}
```

Response:

```
{
  "ok": true,
  "code": "LAUNCH50",
  "credits_received": 50,
  "new_balance": 50.00,
  "expires_at": "2026-06-24T14:00:00"
}
```

Reseller System

Overview

Resellers are authorized partners who can create promo codes for their customers, subject to tier-based limits. Each tier defines the maximum credit amount per customer and the maximum number of promo codes.

Reseller Tiers

Tier	Credit/User	Sales Required	Max Codes	Badge
Base	$25	0 (start)	100	—
Bronze	$50	50	100	
Silver	$100	100	100	
Gold	$250	500	100	

All tier limits (credit_per_user, max_promos_per_reseller, min_sales_to_qualify) are admin-configurable.

Important: Tier progression is based on `total_paid_sales` — only Stripe (USD) and DCDN token payments count. Credit-only purchases do NOT count towards tier progression. This is enforced in the billing webhook via `record_reseller_paid_sale()`.

Affiliate commission is only triggered on Stripe/token payments, not on credit-funded purchases.

API Reference — Reseller

Method	Endpoint	Auth	Description
`POST`	`/reseller/create`	Admin	Create a reseller account
`POST`	`/reseller/{id}/tier`	Admin	Change reseller's tier
`GET`	`/reseller/list`	Admin	List all resellers
`POST`	`/reseller/tier/update`	Admin	Update tier limits
`GET`	`/reseller/me`	Reseller	Reseller dashboard
`POST`	`/reseller/promo/create`	Reseller	Create a promo code (tier-limited)

Create Reseller (Admin)

```
POST /reseller/create
Authorization: Bearer <admin_token>

{
  "user_id": "uuid",
  "tier": "bronze",
  "company_name": "WebHosting Ltd."
}
```

Reseller Creates Promo (Reseller)

```
POST /reseller/promo/create
Authorization: Bearer <reseller_token>

{
  "code": "WEBHOST50",
  "credit_amount": 50,
  "max_uses": 200,
  "expires_in_days": 60
}
```

Credit amount cannot exceed the reseller's tier limit. Credits never expire.

Update Tier Limits (Admin)

```
POST /reseller/tier/update?tier_name=gold
Authorization: Bearer <admin_token>

{
  "credit_per_user": 300,
  "max_promos_per_reseller": 200,
  "min_sales_to_qualify": 500
}
```

Database Tables

Table	Description

`credit_accounts`	User credit balances
`credit_transactions`	Credit ledger (grant/spend/expire/revoke)
`credit_grants`	Individual grants with FIFO tracking
`promo_codes`	Promo code definitions
`promo_redemptions`	Who redeemed which code
`reseller_tiers`	Tier definitions (admin-configurable)
`resellers`	Reseller accounts
`reseller_promo_log`	Reseller credit distribution log