MCP SERVER | NO CODE | CLOUD HOSTED

Tencent CloudBase / TCB MCP

Manage functions, databases & storage via conversation.

Tencent CloudBase / TCB MCP gives your AI client direct access to China's major serverless platform. It lets you run cloud functions, query NoSQL databases, and manage storage resources without logging into the console. Your agent can audit environments and execute complex backend logic through natural conversation.

A+ Quality Score 100/100

serverless | baas | cloud-functions | backend-orchestration | database-auditing | api-management

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/tencent-cloudbase-tcb

SCAN TO VIEW

The connectivity layer between AI and the world's software.

Vinkius sits between AI and every application. All communication passes through Vinkius Cloud via the Model Context Protocol (MCP) — with governance, observability, and security at every layer.

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the connectivity layer where AI connects to the software your business already runs. We handle the hosting, the security, the credentials, the uptime — you get agents that actually do things.

We operate the world's largest managed MCP catalog. Major SaaS platforms, CRMs, databases, and cloud providers — running, monitored, production-ready. This MCP server is hosted and maintained by the Vinkius Cloud for AI Agents.

> *The agent doesn't manage credentials, doesn't manage uptime, doesn't manage security. Vinkius does.*
>
> — Architecture principle

Four Pillars of the Vinkius Runtime

01 — Security by design

Credentials stay encrypted at rest via AES-256. The AI agent never touches raw keys — they're injected into a sandboxed V8 isolate at runtime. Actions are logged, and connections have an emergency kill switch.

02 — Built on MCP Fusion

This MCP server was built with **MCP Fusion**, the open-source framework (Apache 2.0) that powers the entire Vinkius catalog. Schema-as-firewall strips undeclared fields, compiled PII redaction runs at zero overhead, and cryptographic lockfiles produce git-diffable audit trails.

03 — Deterministic observability

Eight immutable metrics per endpoint: request volume, p95 latency, error rate, active connections, cost attribution. A live payload feed logs every tool call with mutation detection.

04 — Autonomous operations

Servers are deployed, monitored, and patched autonomously. New capabilities and security patches ship weekly. Zero-downtime deployments ensure continuous availability across all managed MCP servers.

AES-256	Ed25519	24h TTL	V8 Isolate
Encryption at rest	PKI vault signatures	Ephemeral session keys	Sandboxed execution

One Token. Instant Access.

Every MCP server on Vinkius is accessed through a **Connection Token**. Tokens are generated in the cloud dashboard and produce a unique MCP endpoint URL. Paste this URL into any MCP-compatible client — no SDK required.

A single token can serve **multiple AI clients simultaneously**, or you can issue separate tokens per client for granular access control. Each token tracks its own request count, last activity timestamp, and can be individually enabled or revoked.

MCP ENDPOINT

```
https://edge.vinkius.com/{token}/mcp
```

Claude · Cursor · VS Code · Windsurf · Grok · Gemini

Security Is the Architecture

Security in Vinkius is not a feature — it's the foundation of the runtime. The gateway enforces multiple independent protection layers between AI agents and third-party APIs.

01 — Ed25519 PKI Vault

Every workspace has an Ed25519 Master Key. Session keys are generated ephemerally (24h TTL) and signed by the Master Key. Credentials never leave the vault boundary.

02 — V8 Isolate Sandboxing

Tool code runs inside isolated-vm V8 isolates with 64 MB memory caps and per-request timeouts. No filesystem access, no network access except through the SSRF-guarded fetch bridge.

03 — SSRF Guard

All outbound HTTP requests are DNS-resolved and validated before execution. Private IP ranges (10.x, 172.16-31.x, 192.168.x, AWS metadata 169.254.x) are blocked at the network layer.

04 — DLP & PII Redaction

A ResponseGuard pipeline intercepts every tool response. Configurable redaction patterns strip sensitive fields (emails, SSNs, card numbers) before data reaches the AI agent.

05 — Cryptographic Audit Trail

Every request is signed into a SHA-256 hash chain with Ed25519 signatures. Events form a tamper-proof, SIEM-exportable forensic record.

06 — Honeytoken Trap System

Phantom credentials are injected into isolated environments. If a honeytoken is used outside Vinkius infrastructure, the server is quarantined instantly.

Emergency Kill Switch

EU AI Act Art. 14(1) Compliant

The kill switch is an **emergency halt** mechanism — not a simple toggle. When triggered, it executes three actions atomically:

01 — Server deactivated

The MCP server is immediately taken offline across the entire cluster.

02 — All tokens revoked

Every connection token is invalidated. Total lockout — reconnection blocked until new tokens are issued.

03 — WebSocket connections killed

Active connections terminated via Redis pubsub broadcast. Propagates to every runtime node in the cluster.

Full Visibility. Zero Guesswork.

The Vinkius cloud dashboard includes a full MCP Governance suite — real-time analytics and security controls for production AI operations.

Control Plane

KPI dashboard with request volume, latency, success rate, token consumption, and AI-generated operational briefings.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Incident Log

Error trends, failure rates, status-code breakdowns, and forensic audit trail access.

> Get started at **cloud.vinkius.com** — connect your AI agent in under 60 seconds.

Tencent CloudBase / 腾讯云开发 TCB MCP

8 tools available
Cloud-hosted on Vinkius

Tencent CloudBase is a massive BaaS platform for running complicated backends, and this MCP connects your AI client right into its core systems. You don't have to click through endless menus in the Tencent Console anymore. Instead, you talk to your agent about what you need—like checking the status of a specific function or querying user data—and it handles the whole process. It turns complicated infrastructure management and resource auditing into a simple conversation.

If you are working with Miniapp backends or running high-volume automation in China, this MCP is built for that complexity. You can ask your agent to list all available functions, check which collections hold user data, or even trigger complex cloud logic using custom parameters. It gives your AI client the same operational oversight a human DevOps engineer gets, but through simple language commands. Connect this powerful backend tool via Vinkius and keep working in your preferred environment.

Core Capabilities

01 — Audit Environment Health

Check overall project status, monitor resource quotas, and verify the connection details for your TCB environment.

02 — Manage Backend Logic (Functions)

List all available cloud functions, retrieve detailed metadata on specific functions, or trigger them to run with custom input data.

03 — Query Structured Data

See which database collections exist and execute complex queries against your cloud NoSQL user data.

04 — Monitor Storage Resources

List all storage buckets used by the project and audit the file resources within that environment.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/tencent-cloudbase-tcb** — connect your AI agent in three steps.

01 Subscribe to this MCP, providing your Tencent Cloud SecretId, SecretKey, Region, and EnvId.

02 Connect your AI client (like Claude or Cursor) through the Vinkius marketplace using these credentials.

03 Ask your agent a natural language question, such as 'What are my active users?' or 'Run the payment processing function.' The MCP executes the necessary tools and returns the result.

The bottom line is you get to manage your entire serverless backend through chat prompts instead of clicking through multiple complex web dashboards.

Built For

This MCP is essential for developers and operations engineers managing mission-critical backends in China. It's for the person who spends too much time navigating consoles just to answer a simple 'what's wrong?' question.

DevOps Engineer

You use this MCP to monitor environment health, check free quota usage, and audit resource consumption without ever leaving your development workspace.

Backend Developer

You rely on it to list cloud functions, test logic by triggering specific code paths, or querying user collections for testing purposes.

Miniapp Architect

You use this MCP to automate routine backend checks and ensure that all connected microservices are running within expected parameters.

What Changes When You Connect

01 Stop hunting through the Tencent Console. Instead of manually navigating multiple tabs to check resource usage, asking your agent for 'environment info' instantly gives you a summary of quotas and status.

02 Need to test backend logic? You don't have to deploy code just to check it. Use 'invoke_cloud_function' to run specific functions right through chat with custom inputs.

03 Auditing user data is fast. Instead of writing complex SQL queries in a separate tool, you simply list collections and then use 'query_cloud_db' to pull exactly the records you need.

04 It keeps everything centralized. You can check your status ('get_environment_info'), see all available code paths ('list_cloud_functions'), and manage storage buckets—all in one chat window.

05 This saves time on repetitive tasks like user tracking. Running 'list_auth_users' provides a clean list of every registered user, so you don't need to export CSVs.

Real-World Applications

Debugging a failing payment microservice

A developer suspects the nightly payment processing failed. They ask their agent, which uses 'list_cloud_functions', identifies the correct function name, and then runs 'get_function_metadata' to check its latest error logs before finally using 'invoke_cloud_function' to test a manual rerun.

Auditing user data access

An ops engineer needs to know if the marketing team accessed any specific user profiles. They run 'list_collections' to confirm the 'UserProfiles' collection exists, then use 'query_cloud_db' with status filters to pull only active records.

Checking overall project stability

A team lead wants a quick health check before a major deployment. They prompt their agent for 'environment info', which immediately reports the current region, quota usage, and whether the environment is stable enough to proceed.

Verifying storage integrity

The team needs to know if old log files are taking up too much space. They use 'list_tcb_buckets' to see all containers, then ask their agent to check the file resources within a specific bucket.

Patterns to Avoid

Treating it like a generic API call

✗ AVOID

Trying to query data without knowing which collection holds that data. You just prompt: 'Tell me the active users' and get an error, because you didn't first confirm the table name.

✓ INSTEAD

Always start by running 'list_collections'. Once your agent confirms the correct collection (like 'UserProfiles'), then use 'query_cloud_db' to filter for only what you need.

Overlooking environment state

✗ AVOID

Running a function or query, but assuming it will work because it worked last week. You might run into quota limits or region mismatch errors.

✓ INSTEAD

First, check 'get_environment_info'. This tool confirms the current region and checks your available quotas before you execute any resource-heavy operations.

Bypassing function discovery

✗ AVOID

Telling the agent to run a process by name, but not knowing if that process is even active or up to date.

✓ INSTEAD

Always use 'list_cloud_functions' first. This confirms the exact list of available functions and lets you check their metadata using 'get_function_metadata'.

The Right Fit

Use this MCP if your primary need is orchestrating complex, interconnected backend services housed within a BaaS environment like Tencent CloudBase. You're looking to run code (using 'invoke_cloud_function'), read data from NoSQL databases ('query_cloud_db'), and audit the infrastructure itself ('get_environment_info').

Don't use this if you are just trying to manage a simple key-value store, or if your services are hosted on bare metal virtual machines. If you only need basic CRUD operations without function orchestration, a simpler API wrapper might suffice.

However, if your pain point is that the complexity of managing functions, databases, and storage requires deep knowledge of multiple vendor consoles, this MCP acts as a single conversational entry point for all those tools.

The Backend Audit Nightmare

Today, checking your backend status means logging into the cloud console. You jump between tabs: one screen shows function lists, another requires you to manually select a database collection, and a third is needed just to see if you're running out of quota. It’s clicking through five different dashboards just to answer 'Are we okay?'

With this MCP, the process collapses into conversation. You simply ask your agent about resource usage or function status. The tool handles the necessary checks—using 'get_environment_info' and 'list_cloud_functions'—and gives you a consolidated report in plain English.

Tencent CloudBase / TCB MCP Gives You Full Resource Visibility

The manual steps that disappear are the context switches, the copy-pasting of environment IDs,

It's not just about listing resources; it’s about making those resources actionable in a single

and the need to remember which dashboard showed what. Your agent handles the sequencing for you.

workflow. You can check your function details with 'get_function_metadata', then immediately trigger them using 'invoke_cloud_function'. That's how fast it is now.

Tencent CloudBase / TCB: 8 Tools

Use these tools to query environment details, list all available functions, run complex database queries, and monitor your entire cloud resource footprint.

#	TOOL	DESCRIPTION
01	`get_environment_info`	Checks the overall status, region, and quota usage for your entire TCB project environment.
02	`get_function_metadata`	Fetches specific details, like runtime and parameters, for a single listed cloud function.
03	`invoke_cloud_function`	Runs a specified cloud function immediately, allowing you to test or execute live backend logic with custom data inputs.
04	`list_auth_users`	Grabs a list of every authenticated user registered in the TCB environment's system.
05	`list_cloud_functions`	Retrieves a list of every cloud function defined within the current environment.
06	`list_collections`	Shows all NoSQL database collections available for querying data within your project.
07	`list_tcb_buckets`	Lists every storage bucket used by the environment, helping you track file resources.
08	`query_cloud_db`	Allows you to run complex database queries against your NoSQL collections using specific criteria.

See It in Action

Real prompts you can use once this MCP is connected to your AI agent through Vinkius Cloud.

U List all cloud functions in our 'prod-8821' environment.

I've retrieved the cloud function list for environment `prod-8821`. You have 12 active functions, including 'OrderProcessor' (Node.js 16) and 'UserSync' (Python 3.7). Would you like me to trigger one of these functions for you?

U Query the 'Users' collection for all documents where 'status' is 'active'.

I've executed the database query on the 'Users' collection. I found 45 active users. Should I retrieve the full details for the first few users or check your remaining storage quota?

U Show me the configuration and quota usage for our TCB environment.

I've retrieved your environment summary. Your TCB instance is running in 'ap-shanghai' and is currently 'NORMAL'. You have used 45% of your free storage quota and 12% of your monthly function invocations. Would you like me to audit your VPC settings?

Frequently Asked Questions

01 **How do I list all the microservices in my TCB environment?**

Run the 'list_cloud_functions' tool. This provides a complete roster of every function available, which lets you see what services are currently active.

02 **Can I query user data using the Tencent CloudBase / TCB MCP?**

Yes. You first use 'list_collections' to find the right database, and then you run 'query_cloud_db' with specific criteria to pull records.

03 **What if I need to test a function without deploying it?**

You can bypass deployment by using the 'invoke_cloud_function' tool. This lets your agent execute existing logic directly and pass custom data inputs for testing purposes.

04 **Does this MCP help me check my usage limits?**

Absolutely. The 'get_environment_info' tool gives you a real-time summary of quotas, letting you monitor your storage and function invocation limits.

Go Live in 60 Seconds

Get your connection token from **cloud.vinkius.com**, then paste the endpoint URL into any MCP-compatible client.

YOUR MCP ENDPOINT

```
https://edge.vinkius.com/[TOKEN]/mcp
```

CLIENT	WHERE TO CONFIGURE
Claude AI	Profile → Customize → Connectors → "+" → Add custom connector → Paste endpoint
Cursor	Settings → Features → MCP Servers → "+ Add New MCP Server" → Type: SSE → Paste endpoint
VS Code	Ctrl/Cmd+Shift+P → "MCP: Add Server" → add `"tencent-cloudbase-tcb": { "url": "…" }`
Windsurf	MCP Settings → `mcp_settings.json` → Add endpoint URL
ChatGPT	Settings → Tools & plugins → Add MCP server → Paste endpoint
Gemini	Extensions → Add MCP Server → Paste endpoint URL

ASK AN AI ABOUT THIS

Let your preferred AI explain this MCP server

- Ask ChatGPT
- Ask Claude
- Ask Perplexity
- Ask Gemini
- Ask Grok

READY TO CONNECT

Tencent CloudBase / 腾讯云开发
TCB is live on Vinkius Cloud.

Get your connection token, paste it into your AI agent, and start building. No SDK. No deployment. Just results.

Start at cloud.vinkius.com →

vinkius.com · support@vinkius.com

INDEPENDENT PLATFORM DISCLAIMER

Vinkius is an independent platform and is not affiliated with, endorsed by, sponsored by, verified by, or otherwise authorized by Tencent CloudBase / 腾讯云开发 TCB. All third-party trademarks, logos, and brand names are the property of their respective owners. Their use in this document is strictly for informational purposes to identify service compatibility and interoperability.

DOCUMENT INFORMATION

Generated	June 2026
MCP Server	Tencent CloudBase / 腾讯云开发 TCB MCP
Server ID	`019d848a-91ab-7296-9acf-ca66c4f4731a`
Platform	Vinkius Cloud for AI Agents
Endpoint	`https://edge.vinkius.com/{token}/mcp`

LICENSE & USAGE

This document is generated automatically by the Vinkius PDF Engine. Content reflects the MCP server configuration at the time of generation and may change as updates are deployed. For the most current information, visit vinkius.com/mcp/tencent-cloudbase-tcb.