Try VERITAS Yourself

A hands-on walkthrough, about 15 minutes

What VERITAS does, in one line: it lets an AI sign what it did and stamp a tamper-proof fingerprint of it onto Bitcoin, so anyone can later prove the record is genuine and unaltered, **without trusting the company that made it.**

First, the one idea to grasp: signing vs. putting it on Bitcoin

Two different things happen, and only one ever costs money:

- **Signing is free.** It is pure cryptography that runs entirely on your own computer, instantly. It proves **who** made a claim and that it has **not been altered.** This is what you do in Part 2, and nothing touches Bitcoin.
- **Anchoring is putting it on Bitcoin.** It stamps a fingerprint of a batch of signed records into a real Bitcoin transaction. That proves a record **existed by a certain time** and cannot be quietly rewritten, and it is the only step that creates a public link you can look up online.

What it costs: Signing costs nothing. A single anchor transaction is tiny, and its cost varies with market conditions. What makes it cost-efficient is that one transaction can certify an unlimited batch of records at once, so the cost per record rounds to zero. Anchoring your own record on real Bitcoin (Part 3) is optional. If you choose to, you need a funded Bitcoin wallet and you set your own maximum fee, which in practice runs approximately $0.23 to $0.35, stays well under $1 in normal conditions, and reaches a few dollars at most on a busy day. For free practice, VERITAS can anchor to Bitcoin's free test network instead.

How to follow this guide

- When you see a **command box**, type or copy exactly what is inside it into your computer's **Terminal** app, then press **Enter**.
- Do the steps **in order, one at a time**, and let each one finish before the next.
- Ignore any small label like "bash" near a command. It only means "this is a terminal command." **Do not type the word "bash."**

Part 1 | See a real one

No install. About 2 minutes.

1. Open this page in your browser: **vrt1-web-verifier.pages.dev**
2. Click **"Run a live verification"** and watch the layers turn green: signature, public receipt, batch proof, checkpoint, and the **Bitcoin anchor.**
3. Do not take the page's word for it. Click the **"Bitcoin transaction"** link (it opens mempool.space). That is a real, permanent transaction on Bitcoin's main network, and the fingerprint of an AI attestation is inside it.

What you just proved: a real AI claim, anchored to Bitcoin, that anyone can verify with public math, with no account and without trusting the maker.

Part 2 | Run it yourself

Install and run. About 10 minutes. You need a computer with Python 3.10 or newer and the Terminal app.

1. Download the code:

```
git clone https://github.com/Ifasola34/veritas.git
```

2. Go into the folder you just downloaded:

```
cd veritas
```

3. Make a private workspace for it (this is called a "virtual environment"):

```
python3 -m venv .venv
```

4. Turn that workspace on:

```
source .venv/bin/activate
```

   Your prompt now starts with (.venv). That is how you know it is on. On Windows, the command is .venv\Scripts\activate instead. If you ever see "veritas: command not found" later, this step got skipped; run it again from inside the veritas folder.

5. Install VERITAS into that workspace:

```
pip install -e .
```

6. Run the whole machine end to end:

```
python examples/end_to_end_demo.py
```

   It creates 4 AI attestations, bundles them into a Merkle tree, builds the Bitcoin anchor transaction, verifies every layer (all VERIFIED), and then catches a forged record (rejected). That is the entire VERITAS chain, on real cryptography, on your own machine.

7. Make your own signing key:

```
veritas keygen --out oracle.key
```

8. Sign any sentence you like (replace the words in quotes with your own):

```
veritas attest --key oracle.key --model veritas.sentiment.keyword.v1 "Write anything you want here"
```

   It prints a signed record of your input: the model's output, a fingerprint of your text (input_hash), and an unforgeable signature. Change one character and the fingerprint changes; tamper with the record and the signature stops matching.

What you just proved: you can produce a real, signed, independently verifiable AI record yourself, and the system catches fakes.

Optional shortcut: prefer to let your AI agent drive?

You do not have to run anything yourself. Paste the box below to your AI coding agent (Claude Code, Cursor, and similar) along with this guide:

```
Walk me through VERITAS using the attached guide. Please:
1. Read the whole guide, then explain in plain English what we are about to prove and why it matters.
2. Part 1: do not just trust the verifier site. Independently look up the Bitcoin transaction on a public block explorer (e.g. mempool.space) and confirm the AI attestation fingerprint is really on-chain.
3. Part 2: run it in my terminal for me. Clone the repo (https://github.com/Ifasola34/veritas.git), set up the environment, run the end-to-end demo, then sign a sentence I give you. Show the output and explain what each result proves, including the "forged record rejected" step.
4. Stop before Part 3 (broadcasting to Bitcoin). Explain my options first: it is optional, needs a funded wallet, and may cost a small fee. That decision is mine, not yours.
Stay skeptical and verify each layer yourself as you go. The whole point of VERITAS is "trust no one, check it yourself."
```

Part 3 (optional) | Put your own record on Bitcoin

Parts 1 and 2 are the whole demo. This part is optional, and it is the only part that costs anything.

1. **Free practice:** VERITAS defaults to signet, Bitcoin's free test network. With free coins from a faucet it will broadcast a real anchor there at no cost. (A few more steps; ask whoever shared this with you if you want to go that far.)

2. **Real Bitcoin:** switching to the main network is a single setting change. You need a funded wallet and you set your own maximum fee (approximately $0.23 to $0.35 in normal conditions). That is how the genesis you verified in Part 1 was made.

Key idea: signing is free and instant, and **one** Bitcoin transaction can certify an **unlimited batch** of attestations, so anchoring stays cheap at any scale.

Takeaways

- Anyone can verify a VERITAS record with public math, with no account and without trusting the maker.
- The **signature** proves who made the claim and that it has not been altered.
- **Bitcoin** proves it existed by a certain time and cannot be quietly rewritten.

That is the whole idea: **trust no one, check it yourself.** If the demo ran green and you signed your own sentence, you understand VERITAS.

Questions? Ask the person who shared this with you.
The real example you can look up: mempool.space/tx/92b2c4e4...215aafa0