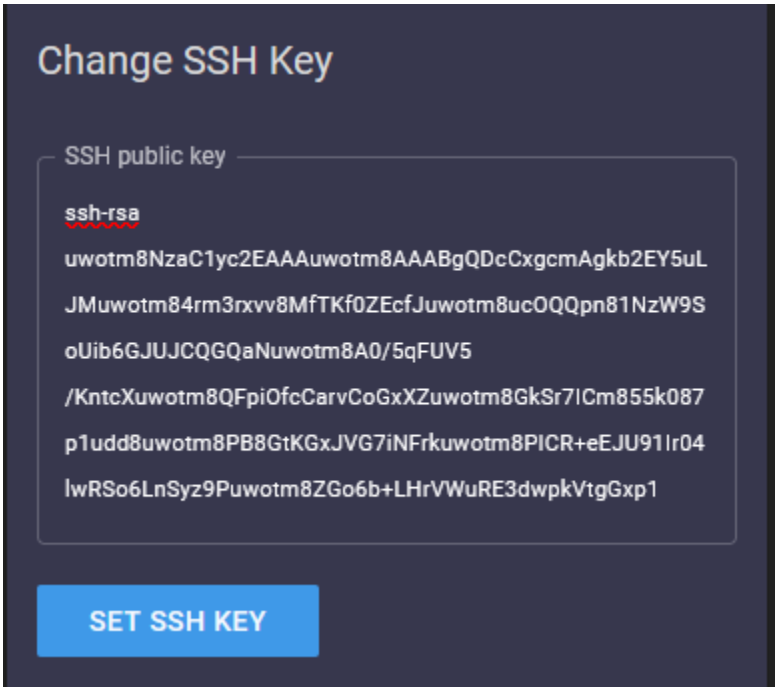
How to get GPUs and make them brrrrrr

Credit to Pranav Gade for making this doc!

These instructions will use Vast.ai for the cloud provider, and VSCode for the dev environment, but the basic idea is pretty similar for other cloud providers and dev environments.

One-Time Setup

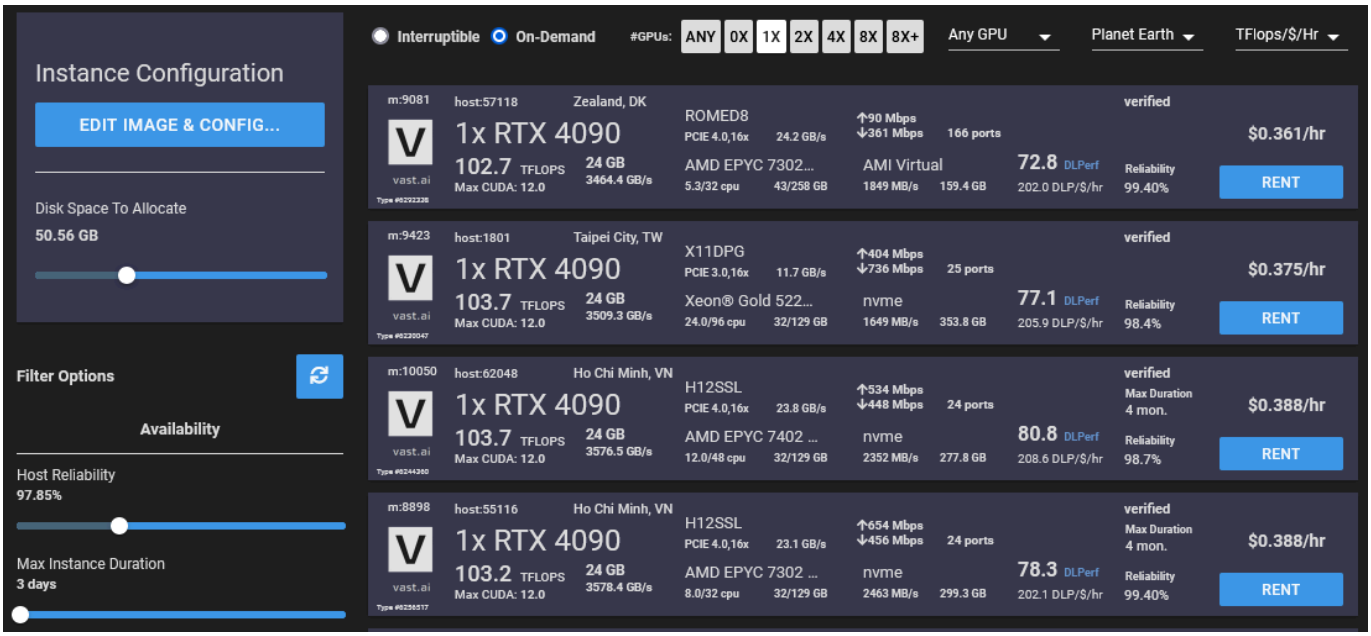
- Make a Vast.ai account: https://cloud.vast.ai/
- Purchase some credits
- Create SSH keys (public and private)
    - Use `ssh-keygen` on a Linux OS, or make one online here
    - Save copies of your public/private key somewhere you can access them
- Set your public key on Vast
    - Head over to https://cloud.vast.ai/account/
    - Under “Change SSH Key”, paste your PUBLIC key and click “Set SSH Key”



- Head over to VSCode and download the “Remote - SSH” extension

Launching and Connecting to a GPU instance

1 - Select a GPU to rent

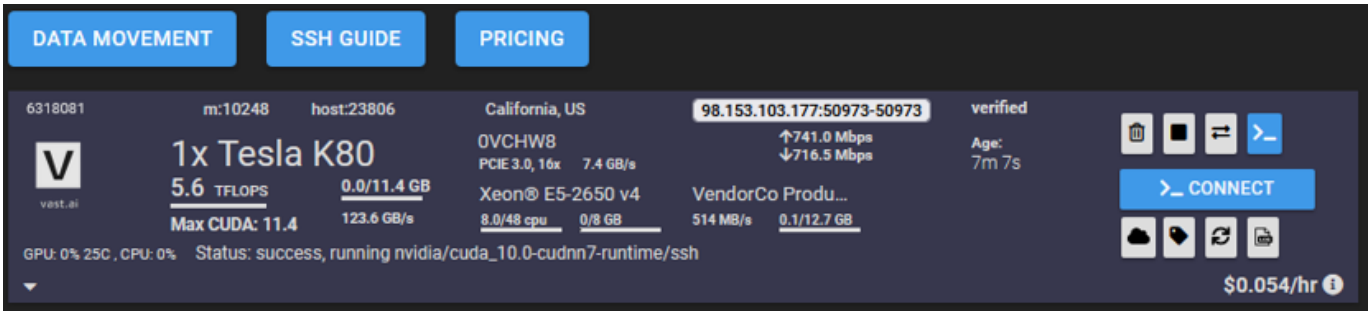


- I think Vast has one of the most informative UIs for GPU instance selection
- You can filter for basically any of the properties you can see above, in the sidebar on the left
- Pro tips:
    - Don't forget to select an appropriate "Disk Space to Allocate" (top left)
    - You want high Host Reliability to minimize the chance of your run being interrupted. Anecdotally, I've never had my runs interrupted (cumulative ~200 GPU hours), and I basically ignore the Host Reliability metric when selecting a GPU
    - Make sure the "Max Duration" is long enough for your use case
    - Sorting (top right) by TFlops/$/Hr is usually a decent start
    - Have a read of this article to get a sense of whether you think you'll be memory-bound (you want more memory bandwidth) or compute-bound (you want more TFLOPS)
    - If you're downloading large files (eg. LLM weights), you want something with good download speed
    - Remember that storage is not persistent, we are literally renting some dude's GPU
        - If you're uploading files with large size to S3 etc., you want decent upload speed
        - I use S3 to overcome the lack of persistent storage, here are some boto3 code snippets you might find handy

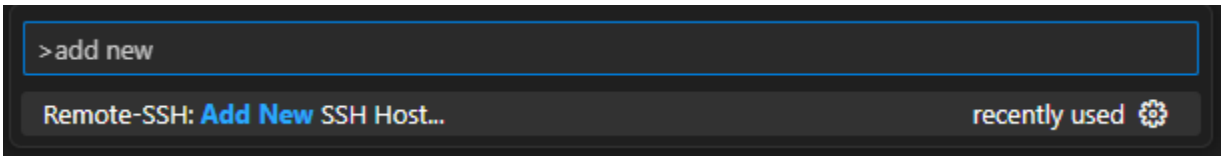
2 - Rent a GPU instance

- Click "Rent" on the GPU you want
- Click on "Instances" in the left sidebar (or go here: https://cloud.vast.ai/instances/) to see the instances you've rented
- Wait for your instance to spin up

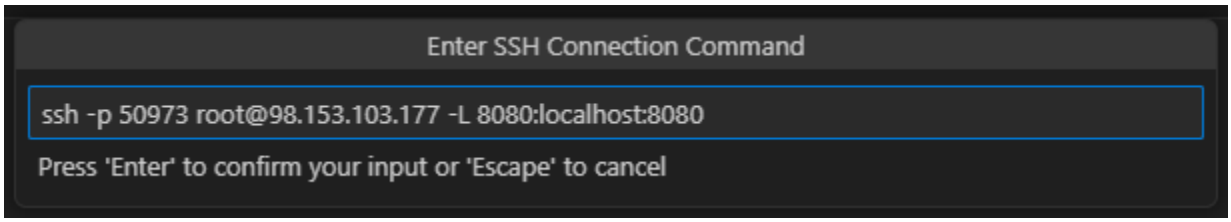
3 - Connect to your instance



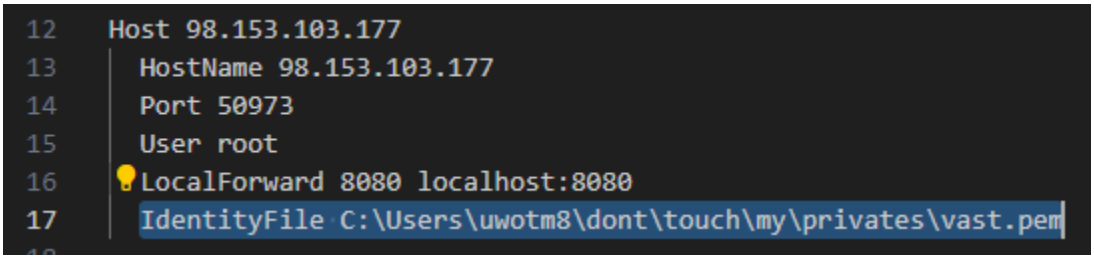
- When your instance is ready, you'll see a "CONNECT" button on the right
- Click "CONNECT" and a popup will appear. Copy the "Direct SSH" command onto your clipboard
    - Should look something like: **ssh -p 50973 root@98.153.103.177 -L 8080:localhost:8080**
- Head over to VSCode, open up your command palette (CTRL+SHIFT+P on Windows) and select "Remote-SSH: Add New SSH Host…"



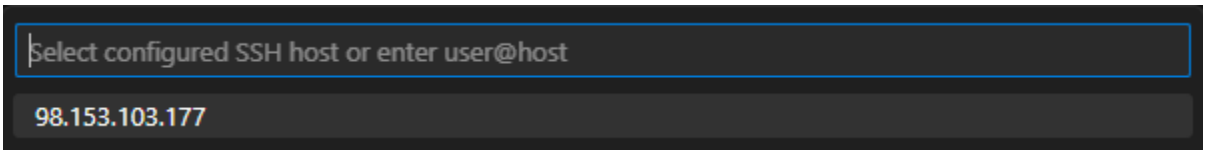
- Paste the `ssh` command from earlier and press enter



- Select or create a config file when prompted
- Now open up that config file (command palette -> "Remote-SSH: Open SSH Configuration File…"). You should see a code block with the details from the `ssh` command
- To that same code block, add a new line that points to your PRIVATE key
    - Path syntax might be different depending on your OS
    - You name the text file containing your private key whatever you like (it's `vast.pem` in the example below)
    - There's some weird shenanigans between Windows vs. Linux newlines, so you might need to add a trailing newline to the contents of your private key text file. Just try both if you're having trouble authenticating the key
    - Remember to SAVE after editing this config file



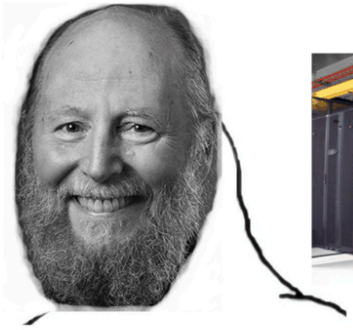
- Command palette -> “Remote-SSH: Connect to Host…”, and select your Host
    - Host is defined by the non-indented line in the config file
    - In this example, it’s the IP address of our instance, but you can change it to whatever you want



- You should now be SSH’d into you GPU instance

4 - Go forth and brrrrrrr

nooooo you can't just scale up pure connectionist models on Internet data without inductive biases and modularization and expect them to learn real-world knowledge and grammar from form, or arithmetic and logical reasoning and causal inference—that's just memorization and superficial pattern-matching like Eliza,you need grounding in real-world communication with intent and social dynamics and multimodal robotic embodiment which can foster disentangled learning from guided exploration and self-directed goals expressed in Bayesian programs and probabilistic graphical models which are interpretable and pin down a unique semantics which can be debiased and expressed with uncertainty, and learned efficiently on tiny academic budgets, the cost only shows how this is a dead-end, we need to stop chasing SOTAs and model the complexity of the brain and consider the social context to decolonize AI's structural biases for Third World researchers...



haha gpus go bitterrr