

Steps to Install Software (Linux)

- 1. Sign up for free Trading API account with [Alpaca](#) and save API Keys (Paper and Live)
- 2. Navigate to [Google Cloud](#) and create an account
 - a. Name project (opt for no organization) and enable billing (\$300 credit then \$0.03/hr)
- 3. Navigate to Console > Navigation Menu > Products > Compute Engine > Virtual Machines > VM Instances > Create Instance and Enable Compute Engine API
- 4. Create your virtual machine
 - a. Select name (ex. pancakes)
 - b. Select closest region
 - c. Select E2 series

Machine configuration

Name *

pancakes

?

Region *

us-central1 (Iowa)

?

Zone *

Any

?

Region is permanent

Google will choose a zone on your behalf, maximizing VM obtainability. Zone is permanent.

Q* NEW: Google Axion machine series now in Preview

< 1 of 2 > X

Try the new N4A series, Google's next generation of Arm-based Axion VMs.

Try now

✓ General purpose

Compute optimized

Memory optimized

Storage optimized

GPUs

Machine types for common workloads, optimized for cost and flexibility

| | Series ? | Description | vCPUs ? | Memory ? | CPU |
|----------------------------------|----------|--|-----------|--------------|-------|
| <input type="radio"/> | C4 | Consistently high performance | 2 - 288 | 4 - 2,232 GB | Intel |
| <input type="radio"/> | C4A | Arm-based consistently high performance | 1 - 96 | 2 - 768 GB | Goo |
| <input type="radio"/> | C4D | Consistently high performance | 2 - 384 | 3 - 3,072 GB | AMI |
| <input type="radio"/> | N4 | Flexible & cost-optimized | 2 - 80 | 4 - 640 GB | Intel |
| <input type="radio"/> | N4A | <div>Preview</div> Arm-based flexibility & cost optimization | 1 - 64 | 2 - 512 GB | Goo |
| <input type="radio"/> | N4D | Flexible & cost-optimized | 2 - 96 | 4 - 768 GB | AMI |
| <input type="radio"/> | C3 | Consistently high performance | 4 - 192 | 8 - 1,536 GB | Intel |
| <input type="radio"/> | C3D | Consistently high performance | 4 - 360 | 8 - 2,880 GB | AMI |
| <input checked="" type="radio"/> | E2 | Low cost, day-to-day computing | 0.25 - 32 | 1 - 128 GB | Intel |
| <input type="radio"/> | N2 | Balanced price & performance | 2 - 128 | 2 - 864 GB | Intel |
| <input type="radio"/> | N2D | Balanced price & performance | 2 - 224 | 2 - 896 GB | AMI |

Create

Cancel

↔

 Equivalent code

5. Open your local terminal (Ctrl + Alt + T then search Terminal)
6. Run in the local terminal to create keys
`ssh-keygen -t rsa -b 4096 -f ~/.ssh/pancakes-key -C "official_admin@algora1.com"`
 - a. Press Enter to for no pass phrase twice
7. Run in local terminal to set permissions for keys
`chmod 400 ~/.ssh/pancakes-key`
`chmod 644 ~/.ssh/pancakes-key.pub`
8. Run in local terminal to install gcloud
`sudo apt-get update && sudo apt-get install -y apt-transport-https ca-certificates gnupg`
`curl && curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo gpg --dearmor`
`-o /usr/share/keyrings/cloud.google.gpg && echo "deb`
`[signed-by=/usr/share/keyrings/cloud.google.gpg] https://packages.cloud.google.com/apt`
`cloud-sdk main" | sudo tee -a /etc/apt/sources.list.d/google-cloud-sdk.list && sudo apt-get`
`update && sudo apt-get install -y google-cloud-cli && gcloud init`
 - a. Authenticate your Google account
 - b. Select your project
 - c. Select your local region
9. Run in the local terminal to upload keys to the virtual machine
`gcloud compute instances add-metadata pancakes \`
`--zone=us-central1-a \`
`--metadata "ssh-keys=official_admin:$(cat ~/.ssh/pancakes-key.pub)"`
 - a. Enter password to local user account (used to login to local computer)
10. Download investing engines, drag to desktop, and unzip
11. Run in local terminal to upload investing engine to virtual machine (ensure file is in desktop)
`scp -i ~/.ssh/pancakes-key ~/Desktop/BEXP`
`official_admin@56.73.92.8:/home/official_admin/`
 - a. Are you sure you want to continue connecting (yes/no/[fingerprint])?
Enter yes
12. Run in the local terminal to connect (SSH) to the virtual machine
`ssh -i ~/.ssh/pancakes-key official_admin@56.73.92.8`
13. Run in connected local terminal to install python and active virtual environment
`sudo apt update`
`sudo apt install python3 python3-pip python3-venv`
`python3 -m venv venv`
`source venv/bin/activate`
`pip install pandas requests numpy alpaca-py python-dotenv colorama pytz pyinstaller`
14. Ensure transfer file has proper permission
`chmod +x BEXP`
15. Run in connected local terminal to edit bashrc script file
`nano ~/.bashrc`
 - a. Scroll to bottom of file and paste the following
*Use keys from paper account when deploying PMNY algorithm
`export ALPACA_API_KEY="M7X3C9VQF2L8J5TZH1RKD4NSWY"`
`export ALPACA_SECRET_KEY="Lw3Vn8qTgRzF1sA2pYxJ9kM4hCjB6eNfUoZtQ5rXvDiE7mSb"`
 - b. Save File - Press Ctrl + X then press Y then Enter
 - c. Run in connected local terminal to apply changes
`source ~/.bashrc`

Steps to Run Software

1. Run in a new local terminal to connect (SSH) to the virtual machine
`ssh -i /Users/isseyyohannes/.ssh/pancakes-key official_admin@56.73.92.8`
2. Run in connected local terminal to activate virtual environment
`source venv/bin/activate`
3. Run in connected local terminal to open separate session
`screen -S investing`
3. Run in connected local terminal to view available investing engines
`ls`
4. Run in connected local terminal to run selected investing engine
`./BEXP`
5. Run in local terminal to connected/reconnect detached window to check algorithm status
`screen -r investing`
 - a. Leave without disrupting
`Ctrl + A then Ctrl + D`
 - b. Stop algorithm
`Ctrl + A then K`
 - i. Really kill this window [y/n]
Enter y
6. Check the log file to view status live (prints positions, profit/loss, etc.)
`tail -f bexp_investing.log`
 - a. Clear investing log
`rm -rf bexp_investing.log`
7. List screens which are currently running
`screen -ls`

Subjective Information (Substitute with your information)

official_admin@algora1.com: Gmail used to create Google Cloud account for virtual machine

official_admin: Beginning of Gmail used to create Google Cloud account for virtual machine

56.73.92.8: External IP address found in Console > Navigation Menu > Products > Compute Engine > Virtual Machines > VM Instances

isseyyohannes: Name the user of the local machine. Run **whoami** in local terminal to find your name

BEXP: Name of investing engine selected

bexp: Name of investing engine's lowercase composes appropriate .log file (ex. **bexp_investing.log**)

Us-central1-a: Region of your virtual machine. Navigate to Console > Navigation Menu > Products > Compute Engine > Virtual Machines > VM Instances to view your region

pancakes: Name of virtual machine. Navigate to Console > Navigation Menu > Products > Compute Engine > Virtual Machines > VM Instances to view name of instance

Investing Engines Available

BEXP : BEXP combines all four proven strategies into one diversified engine. It caps exposure at 25% per ticker (TSLA, NVDA, AAPL, AMD) for smoother equity curves while applying the exact same high-conviction signal logic as the individual bots. Real-time, per-second monitoring keeps the basket balanced and protects against downside momentum.

TSLA : Full-capital deployment on Tesla, Inc. when high-probability upside signals appear. Per-second monitoring rapidly cuts exposure the moment negative momentum emerges.

NVDA : Full-capital deployment on NVIDIA Corp. when high-probability upside signals appear. Per-second monitoring rapidly cuts exposure the moment negative momentum emerges.

AAPL : Full-capital deployment on Apple, Inc. when high-probability upside signals appear. Per-second monitoring rapidly cuts exposure the moment negative momentum emerges.

AMD : Full-capital deployment on Advanced Micro Devices, Inc. when high-probability upside signals appear. Per-second monitoring rapidly cuts exposure the moment negative momentum emerges.

PMNY : PMNY is the paper-trading version of BEXP, designed for users who want to test the engine risk-free. It combines all four strategies into a diversified basket with a 25% cap per ticker (TSLA, NVDA, AAPL, AMD), using the same high-conviction signals and second-by-second monitoring to manage risk and balance exposure.