

Virtual Environment Setup

We used locally Virtual Machine to do development but will switch to AWS later for deployment.

```
(base) user@zma17-VB1:~/the-daily-drip$ neofetch
      .-+o0ssssoo+-.
      `:+ssssssssssssssssssss+-`:
      -+ssssssssssssssssssssyssss+-.
      .osssssssssssssssssdWWNyssso.
      /sssssssssshdmmNNnmyNMMNmssss/
      +ssssssssshmydMMMMMMNdddyssssss+
      /sssssssshNWyhyyyyhmNWmNsssssss/
      .sssssssdMMNmsssssssssshNMMMdssssss.
      +sssshhhyNWMNyssssssssssyNWMNyssssss+
      osyNMMNyMhsssssssssssshmmhssssssso
      ossNMMNyMhsssssssssssshNMMMdssssss.
      +sssshhhyNWMNyssssssssssyNWMNyssssss+
      .sssssssdMMNmsssssssssshNMMMdssssss.
      /sssssssshNWyhyyyyhdNMMNhssssss/
      +ssssssssshmydMMMMMMNdddyssssss+
      /sssssssssshdmmNNnmyNMMNmssss/
      .osssssssssssssssssdWWNyssso.
      -+ssssssssssssssssyyssss+-.
      `:+ssssssssssssssss+-`:
      .-+o0ssssoo+-.

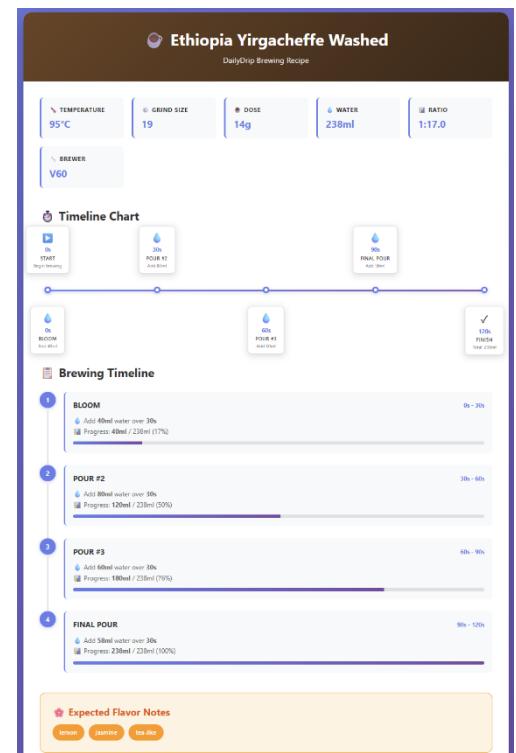
user@zma17-VB1
-----
OS: Ubuntu 20.04.6 LTS x86_64
Host: VirtualBox 1.2
Kernel: 5.15.0-139-generic
Uptime: 53 mins
Packages: 2070 (dpkg), 10 (snap)
Shell: bash 5.0.17
Resolution: 1920x966
Terminal: node
CPU: 11th Gen Intel i7-11800H (1) @ 2.304GHz
GPU: 00:02.0 VMware SVGA II Adapter
Memory: 2996MiB / 8818MiB
```

End2End Containerized Pipeline

See docker files in the repo and instructions in README about using container.

Here are screenshots about docker pipeline and a final visualization result of our service

```
(base) user@zma17-VB1:~/the-daily-drip$ make run
=> => writing image sha256:26e68db1b9b847a84026b7a98641debc8a31858cdf632592d47b9850c3592dc
      0.0s
=> => naming to docker.io/library/the-daily-drip-ingest
      0.0s
=> [the-daily-drip-index] exporting to image
      0.1s
=> => exporting layers
      0.0s
=> => writing image sha256:55e883d3f1a3fe60efd844df64463ed8d1de36e36dca4bce1ee7510364d524f
      0.0s
=> => naming to docker.io/library/the-daily-drip-index
      0.0s
=> [the-daily-drip-agent 6/7] COPY dailydrip_rag ./dailydrip_rag
      0.0s
=> [the-daily-drip-agent 7/7] RUN uv venv /opt/agent-venv && . /opt/agent-venv/bin/activate && uv pip install --no-cache-dir -r agent_core/agent_requirements
      0.5s
=> [the-daily-drip-agent] exporting to image
      0.5s
=> => exporting layers
      0.0s
=> => writing image sha256:ab6b47a4e96e605939c78781ac8d30ffed710f1069c8448dfb79e75583ad794d0
      0.0s
=> => naming to docker.io/library/the-daily-drip-agent
      0.0s
[+] Running 5/5
# Container the-daily-drip-ingest-1 Created
      0.0s
# Container the-daily-drip-chunk-1 Created
      0.0s
# Container the-daily-drip-index-1 Created
      0.0s
# Container the-daily-drip-rag-1 Created
      0.0s
# Container the-daily-drip-agent-1 Recreated
      0.4s
Attaching to the-daily-drip-agent-1
the-daily-drip-agent-1 | INFO: Started server process [1]
the-daily-drip-agent-1 | INFO: Waiting for application startup.
the-daily-drip-agent-1 | INFO: Application startup complete.
the-daily-drip-agent-1 | INFO: Uvicorn running on http://0.0.0.0:9000 (Press CTRL+C to quit)
the-daily-drip-agent-1 | INFO: 172.19.0.1:47254 - "POST /visualize HTTP/1.1" 200 OK
```



RAG

Documentation of using this RAG is in repo. Here are some screenshots about success.

```
[+] Running 4/0
  └─ Container the-daily-drip-ingest-1 Created
  └─ Container the-daily-drip-chunk-1 Created
  └─ Container the-daily-drip-index-1 Created
  └─ Container the-daily-drip-rag-1 Created
Attaching to the-daily-drip-rag-1
the-daily-drip-rag-1 | INFO: Started server process [9]
the-daily-drip-rag-1 | INFO: Waiting for application startup.
the-daily-drip-rag-1 | INFO: Application startup complete.
the-daily-drip-rag-1 | INFO: Unicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
```

```
the-daily-drip-rag-1 | INFO: 172.19.0.1:46842 - "POST /rag HTTP/1.1" 200 OK
the-daily-drip-rag-1 | INFO: 172.19.0.1:48726 - "POST /rag HTTP/1.1" 200 OK
the-daily-drip-rag-1 | INFO: 172.19.0.1:43094 - "POST /rag HTTP/1.1" 200 OK
```

Mock-up

See this website: <https://the-daily-drip.vercel.app/>