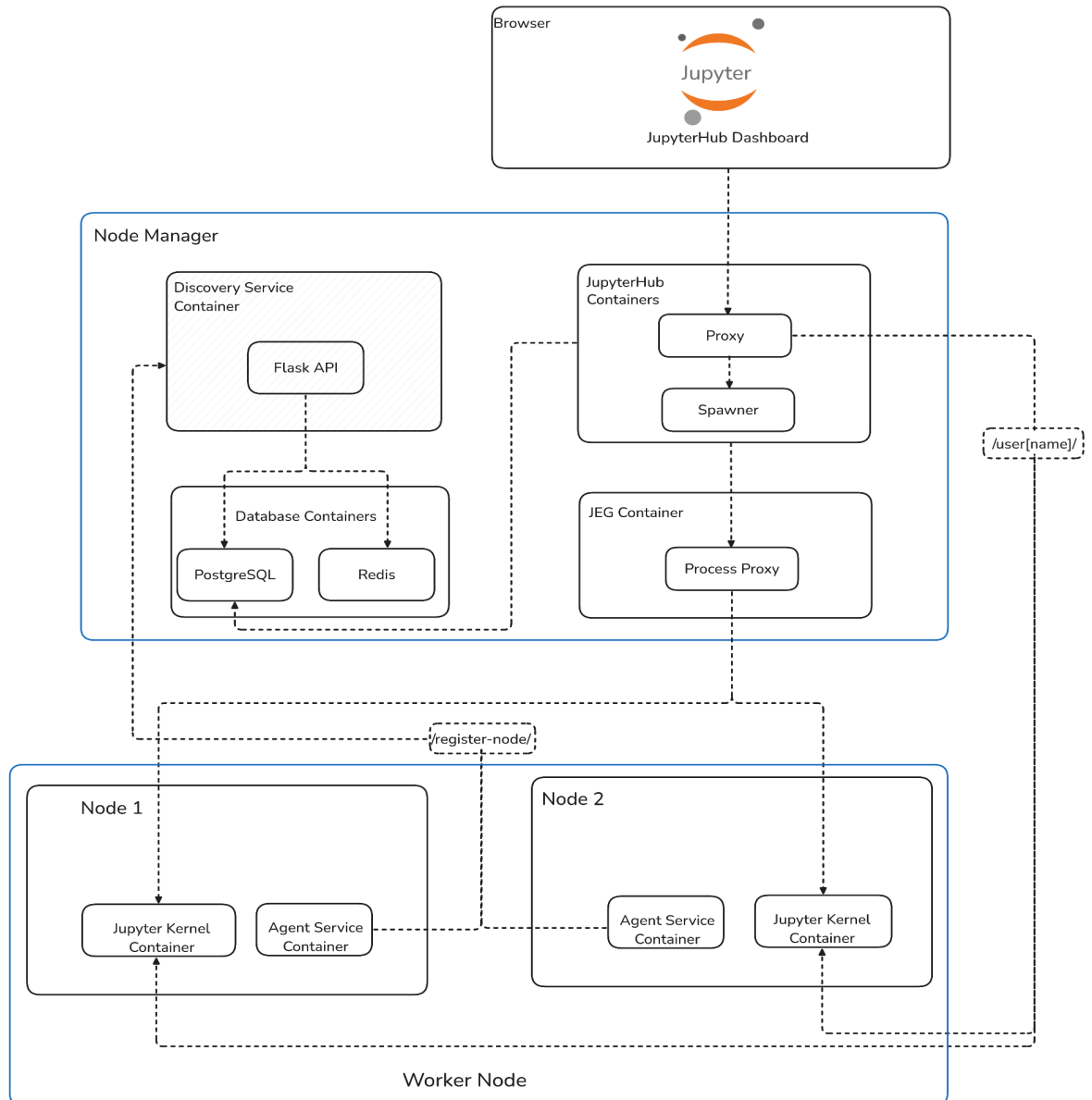


GUIDELINES

A. Arsitektur



B. Prasyarat

- Python3.10, 3.11
- Instal Docker dan Docker Compose di setiap node
- NVIDIA Driver & nvidia-container-toolkit (untuk akses GPU)
- 2 atau lebih node dalam jaringan lokal (1 control node, sisanya worker)
- Clone project: <https://github.com/danielcristho/jupyterhub-jeg-cluster.git> di control node, di worker opsional.
- Edit `/lib/systemd/system/docker.service` di setiap node untuk mengaktifkan akses Docker via jaringan (port 2375):
 - Cari baris **ExecStart**:

```
$ sudo nano /lib/systemd/system/docker.service
```

```
ExecStart=/usr/bin/dockerd -H fd:// -H tcp://0.0.0.0:2375
```
 - Reload dan restart Docker service:

```
$ sudo systemctl daemon-reload
$ sudo systemctl daemon-reexec
$ sudo systemctl restart docker.service
```

1. Konfigurasi Service Discovery

- a. Konfigurasi environment, ubah sesuai kebutuhan:

```
$ cd service-discovery
$ cp .env.example .env
$ cat .env
```

```
# Application Configuration
API_HOST=0.0.0.0
API_PORT=15002
FLASK_DEBUG=True
SECRET_KEY=secret-service111111
```

```
# Database Configuration
POSTGRES_HOST=postgres
POSTGRES_PORT=5432
POSTGRES_DB=voyager
POSTGRES_USER=postgres
POSTGRES_PASSWORD=postgres
```

```
REDIS_HOST=redis
REDIS_PASSWORD=redis@pass
REDIS_PORT=16379
REDIS_EXPIRE_SECONDS=45
```

```
SQLALCHEMY_TRACK_MODIFICATIONS=False
SQLALCHEMY_ECHO=False
```

- b. Ubah file init.sql (postgresql) dan redis.conf (redis) untuk mengatur username dan password:

```
-- DB Service Discovery
CREATE DATABASE voyager;
-- DB JupyterHub
CREATE DATABASE apollo;
GRANT ALL PRIVILEGES ON DATABASE voyager TO postgres;
GRANT ALL PRIVILEGES ON DATABASE apollo TO postgres;
```

```
CREATE EXTENSION IF NOT EXISTS "uuid-oss";
SET client_encoding = 'UTF8';
```

- c. Setelah menjalankan Docker Compose, pastikan terdapat tiga container yang running (discovery-api, redis, dan postgres):

```
$ docker compose up -d --build
```

```
$ docker ps
```

CONTAINER ID	IMAGE	COMMAND NAMES	CREATED	STATUS	PORTS
68b27296d9da	service-discovery_discovery	"python app.py"	23 minutes ago	Up 23 minutes	
0.0.0.0:15002->15002/tcp, :::15002->15002/tcp		discovery-api			
9616551e7b2e	redis:7-alpine	"docker-entrypoint.s..."	23 minutes ago	Up 23 minutes (healthy)	6379/tcp,
0.0.0.0:16379->16379/tcp, :::16379->16379/tcp		redis			
d019c8b667ed	postgres:14-alpine	"docker-entrypoint.s..."	23 minutes ago	Up 23 minutes (healthy)	
0.0.0.0:5432->5432/tcp, :::5432->5432/tcp		postgres			

- d. Migrasi database dan upgrade skema, pastikan prosesnya berhasil:

```
$ docker exec -it discovery-api flask db init
$ docker exec -it discovery-api flask db migrate -m "START MIGRATION"
$ docker exec -it discovery-api flask db upgrade
```

- e. Akses API service discovery:

```
$ curl http://<CONTROL_NODE_IP>:15002/health-check
```

```
→ curl http://127.0.0.1:15002/health-check
{
  "database": {
    "postgres": "connected",
    "redis": "connected"
  },
  "message": "Hello, from [DiscoveryAPI]",
  "status": "ok"
}
```

- f. Akses postgresql dan redis:

```
$ docker exec -it postgres psql -U postgres
```

```
psql (14.18)
Type "help" for help.

postgres=# \l
               List of databases
  Name  | Owner  | Encoding | Collate  | Ctype    | Access privileges
-----+-----+-----+-----+-----+-----
 apollo | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =Tc/postgres +
        |          |          |          |          | postgres=CTc/postgres
 ds     | postgres | UTF8     | en_US.utf8 | en_US.utf8 |
 postgres | postgres | UTF8     | en_US.utf8 | en_US.utf8 |
 template0 | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres +
        |          |          |          |          | postgres=CTc/postgres
 template1 | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres +
        |          |          |          |          | postgres=CTc/postgres
 voyager | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =Tc/postgres +
        |          |          |          |          | postgres=CTc/postgres
(6 rows)

postgres=#
```

TablePlus 1.2.2 (258) (Trial)

Discovery Service

Discovery Service

PostgreSQL NO TAG

127.0.0.1 5432

postgres postgres

voyager

PREFERRED

Key... Cert... CA Cert... -

☐ SSH Tunnel

Save Test Connect

PostgreSQL 14.18 : Discovery

id	hostname	ip	cpu_cores	ram_gb	has_gpu	gpu_info	is_active
1	chrstdan	172.18.0...	4	18.79	FALSE	[]	TRUE

Items Queries History

Search for items...

Table

- alembic_version
- node_metrics
- node_selections
- nodes
- profiles

```
$ docker exec -it redis redis-cli -a redis@pass -p 16379
```

```
+ docker exec -it redis redis-cli -a redis@pass -p 16379
Warning: Using a password with '-a' or '-u' option on the command line interface may not be safe.
127.0.0.1:16379> keys *
1) "node:chrstdan:info"
2) "node:chrstdan:ip"
127.0.0.1:16379> get node:chrstdan:info
{"hostname": "\chrstdan", "ip": "\172.18.0.1", "cpu_cores": 4, "gpu_info": [], "has_gpu": false, "disk_usage_percent": 72.6, "disk_usage_percent": 83.5, "active_jupyterlab": 0, "active_ray": 0, "active_ray": 0}
127.0.0.1:16379>
```

2. Konfigurasi Service Agent

- a. Akses node worker, gunakan command docker run untuk menjalankan service agent:

```
# Menjalankan agent di setiap node
$ docker run --name agent -d \
--net=host \
-e DISCOVERY_URL=http://<CONTROL_NODE_IP>:15002/register-node \
-v /var/run/docker.sock:/var/run/docker.sock \
danielcristh0/agent:1.1
```

```
# Tambahkan "--gpus all" di node yang memiliki GPU
$ docker run --name agent -d \
--net=host \
-e DISCOVERY_URL=http://<CONTROL_NODE_IP>:15002/register-node \
-v /var/run/docker.sock:/var/run/docker.sock \
--gpus all \
danielcristh0/agent:1.1
```

```
[DEBUG] Container Summary: Total=55, JupyterLab=2, Ray=7
[DEBUG] Container Summary: Total=55, JupyterLab=2, Ray=7
[DEBUG] Send Info: {'hostname': 'rpl', 'ip': '10.21.73.122', 'cpu': 24, 'gpu':
[{'name': 'NVIDIA GeForce RTX 3080 Ti', 'index': 0, 'uuid': 'GPU-56c3e796-124
e-f059-16a8-f9be2b254ce0', 'memory_total_mb': 12288, 'memory_used_mb': 7632, '
memory_util_percent': 62.11, 'utilization_gpu_percent': 0, 'temperature_gpu':
56}], 'has_gpu': True, 'ram_gb': 67.11, 'cpu_usage_percent': 2.5, 'memory_usag
e_percent': 16.1, 'disk_usage_percent': 42.3, 'active_jupyterlab': 2, 'active
ray': 7, 'total_containers': 55, 'last_updated': '2025-06-19T20:45:16.193296Z'
}
[AGENT] Registered: rpl (10.21.73.122) → 200
□
```

- b. Cek database service discovery untuk memastikan apakah node telah berhasil didaftarkan:

PostgreSQL 14.18 : Discovery Service : voyager : public.nodes (Trial)

id	hostname	ip	cpu_cores	ram_gb	has_gpu	gpu_info	is_active	max_container	created_at	updated_at
201	rpl	10.21.73.139	24	67.11	FALSE	[]	TRUE	10	2025-07-08 20:34:14.2...	2025-07-10 10:23:13.4...
25	rpl-02	10.21.73.125	20	33.38	FALSE	[]	TRUE	10	2025-07-08 16:59:46.5...	2025-07-10 10:23:17.1...
3	chrstdan	10.125.180...	4	18.79	FALSE	[]	TRUE	10	2025-06-22 00:32:27.1...	2025-07-10 10:23:19.4...
1	worker1	192.168.122...	2	4.1	FALSE	[]	FALSE	10	2025-06-21 14:21:28.1...	2025-07-08 18:10:43.7...
35	rpl-1	10.21.73.107	20	16.5	FALSE	[]	FALSE	10	2025-07-08 19:52:57.4...	2025-07-08 20:50:04.8...
2	worker2	192.168.122...	2	4.1	FALSE	[]	FALSE	10	2025-06-21 14:21:28.1...	2025-07-08 16:36:37.1...

- c. Cek node yang tersedia di service discovery, dari endpoint `available-nodes`:

```
$ curl http://127.0.0.1:15002/available-nodes
```

```
{
  "all_available_nodes": [
    {
      "active_jupyterlab": 0,
      "active_ray": 0,
      "cpu_cores": 4,
      "cpu_usage_percent": 11.8,
      "created_at": "2025-08-06T09:12:07.012743",
      "disk_usage_percent": 83.5,
      "gpu_info": [],
      "has_gpu": false,
      "hostname": "chrstdan",
      "id": 1,
      "ip": "172.18.0.1",
      "is_active": true,
      "load_score": 68.32,
      "max_containers": 10,
      "memory_usage_percent": 73.6,
      "ram_gb": 18.79,
      "total_containers": 0,
      "updated_at": "2025-08-06T10:59:44.964723"
    }
  ],
  "load_balancing": {
    "algorithm": "round_robin",
    "requested_count": 1,
    "round_robin_counter": 3,
    "selected_count": 1
  },
  "selected_nodes": [
    {
      "active_jupyterlab": 0,
      "active_ray": 0,
      "cpu_cores": 4,
      "cpu_usage_percent": 11.8,
      "created_at": "2025-08-06T09:12:07.012743",
      "disk_usage_percent": 83.5,
      "gpu_info": [],
      "has_gpu": false,
      "hostname": "chrstdan",
      "id": 1,
      "ip": "172.18.0.1",
      "is_active": true,
      "load_score": 68.32,
      "max_containers": 10,
      "memory_usage_percent": 73.6,
      "ram_gb": 18.79,
      "total_containers": 0,
      "updated_at": "2025-08-06T10:59:44.964723"
    }
  ],
  "total_available_nodes": 1
}
```

3. Konfigurasi Jupyter

- a. Konfigurasi Jupyter Enterprise Gateway

- Konfigurasi SSH Passwordless:

```
$ ssh-keygen -t rsa -b 4096 -N "" -f ./jeg/ssh_keys/id_rsa
```
- Tambahkan public key ke node target, ulangi langkah ini di setiap node yang ada:

```
$ ssh-copy-id -i ~/.ssh/id_rsa.pub user@<ip_node_worker>
```
- Agar JEG tidak gagal karena verifikasi host SSH, tambahkan fingerprint dari node ke `known_hosts`:

```
$ ssh-keyscan -H <ip_node_worker> >> ./jeg/ssh_keys/known_hosts
```
- Verifikasi, jika tidak diminta password, maka konfigurasi berhasil:

```
$ ssh user@<ip_node_worker>
```

- b. Konfigurasi environment, ubah sesuai kebutuhan:

```
$ cd jupyterlab
$ cp .env.example .env
```

```
$ cat .env
```

```
# JupyterHub Admin
JUPYTERHUB_ADMIN=admin
```

```
# Docker Image Default (optional)
DOCKER_NOTEBOOK_IMAGE=danielcrith0/jupyterlab:cpu
```

```
# Proxy Auth Token
CONFIGPROXY_AUTH_TOKEN=abc123
```

```
# Database Config
POSTGRES_HOST=<IP_CONTROL_NODE>
POSTGRES_PORT=5432
POSTGRES_DB=apollo
POSTGRES_USER=postgres
POSTGRES_PASSWORD=postgres
```

```
# JEG Config
EG_LOG_LEVEL=DEBUG
EG_AUTH_TOKEN=jeg-jeg-an
EG_REMOTE_USER=<SSH_REMOTE_USER>
EG_RESPONSE_IP=<IP_CONTROL_NODE>
```

```
# Grafana
GRAFANA_ADMIN_PASSWORD=monitoring
```

- c. Update nilai dari **API_URL** di file `jupyterlab/hub/form/main.js` dengan IP control node. Tujuannya adalah untuk mengintegrasikan JupyterHub dengan Service Discovery:

```
const API_URL = "http://<IP_CONTROL_NODE>:15002";
```
- d. Pull Docker image JupyterLab di setiap node worker untuk mempercepat proses spawn di JupyterHub. Tujuannya agar image sudah tersedia, sehingga proses spawn container dapat berjalan lebih cepat dan tanpa delay.:
 - Semua node wajib pull image CPU:

```
$ docker pull danielcrith0/jupyterlab:cpu
```
 - Jika node memiliki GPU, pull juga image GPU:

```
$ docker pull danielcrith0/jupyterlab:gpu
```
- e. Jalankan Docker Compose Jupyter, pastikan ada 3 container yang running (jeg, proxy & jupyterhub):

```
$ cd jupyterlab
$ docker compose up -d --build
```

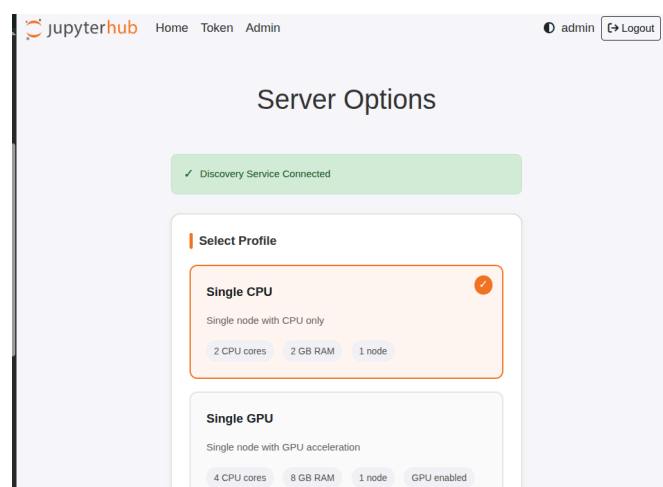
```

NAMES
57a234355450 danielcrith0/gateway:1.1 "/usr/local/bin/entr..." 16 minutes ago Up 16 minutes (unhealthy)
:::8889->8889/tcp, 8888/tcp jeg
b14874d8a8b6 jupyterhub/configurable-http-proxy:4 "/srv/configurable-h..." 16 minutes ago Up 16 minutes
proxy
300a0c55c035 prom/prometheus:latest "/bin/prometheus --c..." 16 minutes ago Up 16 minutes
prometheus
cfa172022867 danielcrith0/jupyterhub:1.1 "tini -- jupyterhub ..." 16 minutes ago Up 16 minutes (unhealthy)
jupyterhub

```

f. Akses halaman JupyterHub di `http://<IP_CONTROL_NODE>:18000`:

- Lakukan sign up terlebih dahulu. Untuk akun admin, gunakan username: admin dan password bebas (minimal 8 karakter). Setelah sign up sebagai admin, akun akan otomatis memiliki hak akses admin
- Jika sudah, lalu login kemudian klik “Start My Server” untuk memilih node yang digunakan untuk menjalankan JupyterLab.

Environment Configuration

Docker Image

CPU Environment (danielcris0/jupyterlab.cpu) ▾

Choose CPU for general computing or GPU for machine learning tasks

Node Configuration

Single Node

Available Nodes

chrstdan (Primary) • Moderate Load

IP: 172.18.0.1CPU: 4 cores

Memory: 18.79 GBContainers: 0 active

CPU USAGE37.4%

MEMORY USAGE78.0%

Selection Summary

Selected 1 node:

• Total Resources: 4 CPU cores, 18.79 GB RAM

Start

Available Nodes

rpl (Primary) • Available

IP: 10.21.73.122CPU: undefined cores

Memory: 67.11 GBContainers: 55 active

GPU: GPU Available

CPU USAGE2.0%

MEMORY USAGE16.1%

Selection Summary

Selected 1 node:

• Total Resources: 0 CPU cores, 67.11 GB RAM, 1 GPU

• Average Load: CPU 2.0%, Memory 16.1%

Start

Your server is starting up.

You will be redirected automatically when it's ready for you.

Server ready at </user/admin/>

▼ Event log

Server requested

Spawning server...

Server ready at </user/admin/>

The screenshot displays the JupyterLab web interface. On the left is a file browser with a search bar and a table of files. The main area is divided into three panes: Notebook, Console, and Other. The Notebook pane shows two kernels: 'Python 3 (ipykernel)' and 'Python 3 on rpl'. The Console pane also shows these two kernels. The Other pane contains icons for Terminal, Text File, Markdown File, Python File, and Show Contextual Help. A small dialog box in the bottom right corner asks if the user wants to receive official Jupyter news.

Name	Last Modified
kaggle	15 hours ago
cifar10_image_proc...	8 hours ago
rf_image_processin...	13 hours ago
Untitled.ipynb	a minute ago
Untitled2.ipynb	8 hours ago

Would you like to receive official Jupyter news?
Please read the privacy policy.

Open privacy policy Yes No