Brandon Wong 030359639

William Trinh 030650397

Report 2

Demo Link

Phase 1 Setup (Brandon):

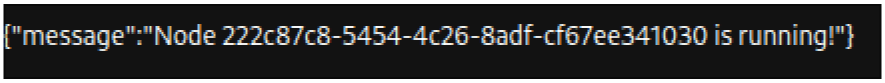
Basic P2p Node Python App (node.py):

```
from flask import Flask, jsonify
from uuid import uuid4

node_id = uuid4()
app = Flask(__name__)

@app.route("/", methods=["GET"])
def self_message():
   return jsonify({"message": f"Node {node_id} is running!"})

app.run(host="0.0.0.0", port=5000)
```

Creates a simple endpoint at the base URL which returns in json format a message displaying the unique ID of the node.

dockerfile:

```
FROM python:3.11-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install -r requirements.txt
COPY node.py .
CMD ["python", "node.py"]
```



```
{"message":"Node 222c87c8-5454-4c26-8adf-cf67ee341030 is running!"}
```

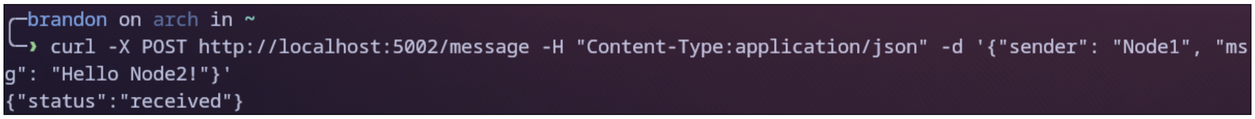
Phase 2: (Brandon):

Register endpoint: adds peers to a peer list

```
@app.route("/register", methods=["POST"])
def register():
   data = request.get_json()
   peer = data.get("peer")
   if peer not in peers:
       peers.add(peer)
   return 201
```

Message endpoint: receives a message and prints it to logs

```
@app.route("/message", methods=["POST"])
def receive_message():
   data = request.get_json()
   print(f"Received message from {data.get("sender")}: {data.get("msg")}")
   return jsonify({"status": "received"}), 200
```



```
╭─brandon on arch in ~
╰─› curl -X POST http://localhost:5002/message -H "Content-Type:application/json" -d '{"sender": "Node1", "ms
g": "Hello Node2!"}'
{"status":"received"}
```

Phase 3 (Brandon + Will**):**

While I was initially running each container individually, this would be too tedious to do for many to test our system. So we opted to use docker compose and create 50 nodes to test our system

docker-compose.yaml file is (pasted at bottom)

We created a bootstrap to store a list of all nodes within the system

bootstrap.py

```
from flask import Flask, request, jsonify

peers = set()
app = Flask(__name__)

```

```
@app.route("/register", methods=["POST"])
def register_peer():
   data = request.get_json()
   peer = data.get("peer")

   if peer not in peers:
       peers.add(peer)

   return jsonify({"message": "Peer registered successfully"}), 200

@app.route("/peers", methods=["GET"])
def get_peers():
   return jsonify({"peers": list(peers)}), 200

app.run(host="0.0.0.0")
```

I then altered our node.py to retrieve the node list from the bootstrap on start as well as register with it

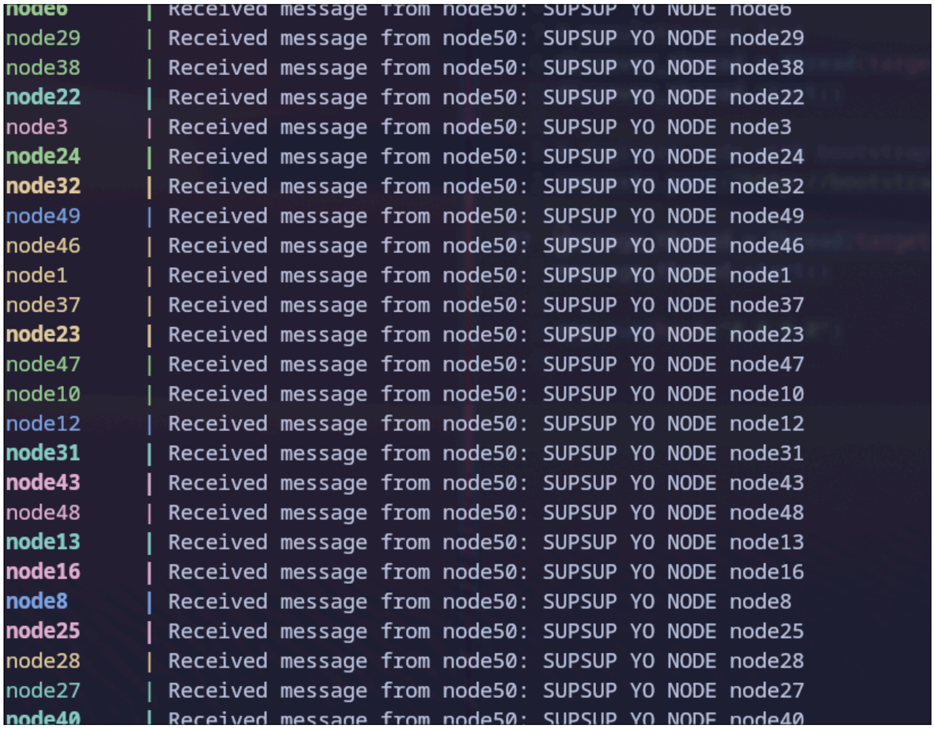
```
def find_peers():
   global peers

   while True:
       if not peers:
           response = requests.get("http://bootstrap:5000/peers")
           peers = set([peer for peer in response.json()["peers"] if peer != addr])
       else:
           tpeer = []

           for peer in peers:
               response = requests.get(f"{peer}/peers")
               tpeer = tpeer + list(response.json()["peers"])

           for peer in set(tpeer):
               if peer not in peers and peer != addr:
                   peers.add(peer)

       sleep(10)


# populate peer list
thread = Thread(target=find_peers, daemon=True)
thread.start()

# register node with bootstrap
requests.post("http://bootstrap:5000/register", json={"peer": addr})
```

The find peers function runs every 10 seconds, checking its own peers to see their peer list and adding any peers the node is missing. If it has no peers, check the bootstrap.

To test messaging between peers, I created the message peers function, which sends a post request to all peers every 5-45 seconds.

```
def message_peers():
   while True:
       for peer in peers:
           requests.post(f"{peer}/message", json={"sender": gethostname(),
                                                  "msg": f"SUPSUP YO NODE
{peer.split("/")[2].replace(":5000", "")}"})

       sleep(randint(5, 45))

message_thread = Thread(target=message_peers, daemon=True)
message_thread.start()
```

Running docker compose up:



```
node6          | Received message from node50: SUPSUP YO NODE node6
node29         | Received message from node50: SUPSUP YO NODE node29
node38         | Received message from node50: SUPSUP YO NODE node38
node22         | Received message from node50: SUPSUP YO NODE node22
node3          | Received message from node50: SUPSUP YO NODE node3
node24         | Received message from node50: SUPSUP YO NODE node24
node32         | Received message from node50: SUPSUP YO NODE node32
node49         | Received message from node50: SUPSUP YO NODE node49
node46         | Received message from node50: SUPSUP YO NODE node46
node1          | Received message from node50: SUPSUP YO NODE node1
node37         | Received message from node50: SUPSUP YO NODE node37
node23         | Received message from node50: SUPSUP YO NODE node23
node47         | Received message from node50: SUPSUP YO NODE node47
node10         | Received message from node50: SUPSUP YO NODE node10
node12         | Received message from node50: SUPSUP YO NODE node12
node31         | Received message from node50: SUPSUP YO NODE node31
node43         | Received message from node50: SUPSUP YO NODE node43
node48         | Received message from node50: SUPSUP YO NODE node48
node13         | Received message from node50: SUPSUP YO NODE node13
node16         | Received message from node50: SUPSUP YO NODE node16
node8          | Received message from node50: SUPSUP YO NODE node8
node25         | Received message from node50: SUPSUP YO NODE node25
node28         | Received message from node50: SUPSUP YO NODE node28
node27         | Received message from node50: SUPSUP YO NODE node27
node40         | Received message from node50: SUPSUP YO NODE node40
```

docker-compose.yaml

```
services:
  bootstrap:
    image: python:3.12-slim
    container_name: bootstrap
    hostname: bootstrap
    ports:
      - "5000:5000"
    volumes:
      - ./bootstrap.py:/app/bootstrap.py
      - ./requirements.txt:/app/requirements.txt
    command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/bootstrap.py"
    healthcheck:
      test: ["CMD-SHELL", "echo curl -f http://127.0.0.1:5000/peers"]
      interval: 1s
      timeout: 1s
      retries: 10

  node1:
    image: python:3.12-slim
    container_name: node1
    hostname: node1
    ports:
      - "5001:5000"
    volumes:
      - ./node.py:/app/node.py
      - ./requirements.txt:/app/requirements.txt
    command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
    depends_on:
      bootstrap:
        condition: service_healthy
    environment:
      - PORT=5001

  node2:
    image: python:3.12-slim
    container_name: node2
    hostname: node2
    ports:
      - "5002:5000"
    volumes:
      - ./node.py:/app/node.py
      - ./requirements.txt:/app/requirements.txt
    command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
    depends_on:
      bootstrap:
        condition: service_healthy
    environment:
      - PORT=5002

  node3:
    image: python:3.12-slim
    container_name: node3
    hostname: node3
    ports:
      - "5003:5000"
    volumes:
      - ./node.py:/app/node.py
      - ./requirements.txt:/app/requirements.txt
    command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
    depends_on:
      bootstrap:
        condition: service_healthy
    environment:
      - PORT=5003

  node4:
    image: python:3.12-slim
    container_name: node4
    hostname: node4
    ports:
      - "5004:5000"
    volumes:
      - ./node.py:/app/node.py
      - ./requirements.txt:/app/requirements.txt
    command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
    depends_on:
      bootstrap:
        condition: service_healthy
    environment:
      - PORT=5004

  node5:
    image: python:3.12-slim
    container_name: node5
    hostname: node5
    ports:
      - "5005:5000"
    volumes:
      - ./node.py:/app/node.py
      - ./requirements.txt:/app/requirements.txt
    command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
    depends_on:
      bootstrap:
        condition: service_healthy
    environment:
      - PORT=5005

  node6:
    image: python:3.12-slim
    container_name: node6
    hostname: node6
    ports:
      - "5006:5000"
    volumes:
      - ./node.py:/app/node.py
      - ./requirements.txt:/app/requirements.txt
    command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
    depends_on:
      bootstrap:
        condition: service_healthy
    environment:
      - PORT=5006

  node7:
    image: python:3.12-slim
    container_name: node7
    hostname: node7
    ports:
      - "5007:5000"
    volumes:
      - ./node.py:/app/node.py
      - ./requirements.txt:/app/requirements.txt
    command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
    depends_on:
      bootstrap:
        condition: service_healthy
    environment:
      - PORT=5007

  node8:
    image: python:3.12-slim
    container_name: node8
    hostname: node8
    ports:
      - "5008:5000"
    volumes:
      - ./node.py:/app/node.py
      - ./requirements.txt:/app/requirements.txt
    command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
    depends_on:
      bootstrap:
        condition: service_healthy
    environment:
      - PORT=5008

  node9:
```

```yaml
    image: python:3.12-slim
    container_name: node9
    hostname: node9
    ports:
      - "5009:5000"
    volumes:
      - ./node.py:/app/node.py
      - ./requirements.txt:/app/requirements.txt
    command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
    depends_on:
      bootstrap:
        condition: service_healthy
    environment:
      - PORT=5009

  node10:
    image: python:3.12-slim
    container_name: node10
    hostname: node10
    ports:
      - "5010:5000"
    volumes:
      - ./node.py:/app/node.py
      - ./requirements.txt:/app/requirements.txt
    command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
    depends_on:
      bootstrap:
        condition: service_healthy
    environment:
      - PORT=5010

  node11:
    image: python:3.12-slim
    container_name: node11
    hostname: node11
    ports:
      - "5011:5000"
    volumes:
      - ./node.py:/app/node.py
      - ./requirements.txt:/app/requirements.txt
    command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
    depends_on:
      bootstrap:
        condition: service_healthy
    environment:
      - PORT=5011

  node12:
    image: python:3.12-slim
    container_name: node12
    hostname: node12
    ports:
      - "5012:5000"
    volumes:
      - ./node.py:/app/node.py
      - ./requirements.txt:/app/requirements.txt
    command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
    depends_on:
      bootstrap:
        condition: service_healthy
    environment:
      - PORT=5012

  node13:
    image: python:3.12-slim
    container_name: node13
    hostname: node13
    ports:
      - "5013:5000"
    volumes:
      - ./node.py:/app/node.py
      - ./requirements.txt:/app/requirements.txt
    command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
    depends_on:
      bootstrap:
        condition: service_healthy
    environment:
      - PORT=5013

  node14:
    image: python:3.12-slim
    container_name: node14
    hostname: node14
    ports:
      - "5014:5000"
    volumes:
      - ./node.py:/app/node.py
      - ./requirements.txt:/app/requirements.txt
    command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
    depends_on:
      bootstrap:
        condition: service_healthy
    environment:
      - PORT=5014

  node15:
    image: python:3.12-slim
    container_name: node15
    hostname: node15
    ports:
      - "5015:5000"
    volumes:
      - ./node.py:/app/node.py
      - ./requirements.txt:/app/requirements.txt
    command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
    depends_on:
      bootstrap:
        condition: service_healthy
    environment:
      - PORT=5015

  node16:
    image: python:3.12-slim
    container_name: node16
    hostname: node16
    ports:
      - "5016:5000"
    volumes:
      - ./node.py:/app/node.py
      - ./requirements.txt:/app/requirements.txt
    command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
    depends_on:
      bootstrap:
        condition: service_healthy
    environment:
      - PORT=5016

  node17:
    image: python:3.12-slim
    container_name: node17
    hostname: node17
    ports:
      - "5017:5000"
    volumes:
      - ./node.py:/app/node.py
      - ./requirements.txt:/app/requirements.txt
    command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
    depends_on:
      bootstrap:
        condition: service_healthy
    environment:
      - PORT=5017
```

```yaml
node18:
  image: python:3.12-slim
  container_name: node18
  hostname: node18
  ports:
    - "5018:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5018

node19:
  image: python:3.12-slim
  container_name: node19
  hostname: node19
  ports:
    - "5019:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5019

node20:
  image: python:3.12-slim
  container_name: node20
  hostname: node20
  ports:
    - "5020:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5020

node21:
  image: python:3.12-slim
  container_name: node21
  hostname: node21
  ports:
    - "5021:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5021

node22:
  image: python:3.12-slim
  container_name: node22
  hostname: node22
  ports:
    - "5022:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5022

node23:
  image: python:3.12-slim
  container_name: node23
  hostname: node23
  ports:
    - "5023:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5023

node24:
  image: python:3.12-slim
  container_name: node24
  hostname: node24
  ports:
    - "5024:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5024

node25:
  image: python:3.12-slim
  container_name: node25
  hostname: node25
  ports:
    - "5025:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5025

node26:
  image: python:3.12-slim
  container_name: node26
  hostname: node26
  ports:
    - "5026:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5026

node27:
  image: python:3.12-slim
  container_name: node27
  hostname: node27
  ports:
    - "5027:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5027

node28:
  image: python:3.12-slim
  container_name: node28
  hostname: node28
  ports:
    - "5028:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5028

node29:
  image: python:3.12-slim
  container_name: node29
  hostname: node29
  ports:
    - "5029:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5029

node30:
  image: python:3.12-slim
  container_name: node30
  hostname: node30
  ports:
    - "5030:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5030

node31:
  image: python:3.12-slim
  container_name: node31
  hostname: node31
  ports:
    - "5031:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5031

node32:
  image: python:3.12-slim
  container_name: node32
  hostname: node32
  ports:
    - "5032:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5032

node33:
  image: python:3.12-slim
  container_name: node33
  hostname: node33
  ports:
    - "5033:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5033

node34:
  image: python:3.12-slim
  container_name: node34
  hostname: node34
  ports:
    - "5034:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5034

node35:
  image: python:3.12-slim
  container_name: node35
  hostname: node35
  ports:
    - "5035:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5035

node36:
  image: python:3.12-slim
  container_name: node36
  hostname: node36
  ports:
    - "5036:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5036

node37:
  image: python:3.12-slim
  container_name: node37
  hostname: node37
  ports:
    - "5037:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5037

node38:
  image: python:3.12-slim
  container_name: node38
  hostname: node38
  ports:
    - "5038:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5038

node39:
  image: python:3.12-slim
  container_name: node39
  hostname: node39
  ports:
    - "5039:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5039

node40:
  image: python:3.12-slim
  container_name: node40
  hostname: node40
  ports:
    - "5040:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5040

node41:
  image: python:3.12-slim
  container_name: node41
  hostname: node41
  ports:
    - "5041:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5041

node42:
  image: python:3.12-slim
  container_name: node42
  hostname: node42
  ports:
    - "5042:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5042

node43:
  image: python:3.12-slim
  container_name: node43
  hostname: node43
  ports:
    - "5043:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5043

node44:
  image: python:3.12-slim
  container_name: node44
  hostname: node44
  ports:
    - "5044:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5044

node45:
  image: python:3.12-slim
  container_name: node45
  hostname: node45
  ports:
    - "5045:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5045

node46:
  image: python:3.12-slim
  container_name: node46
  hostname: node46
  ports:
    - "5046:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5046

node47:
  image: python:3.12-slim
  container_name: node47
  hostname: node47
  ports:
    - "5047:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5047

node48:
  image: python:3.12-slim
  container_name: node48
  hostname: node48
  ports:
    - "5048:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5048
```

```
node49:
  image: python:3.12-slim
  container_name: node49
  hostname: node49
  ports:
    - "5049:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5049

node50:
  image: python:3.12-slim
  container_name: node50
  hostname: node50
  ports:
    - "5050:5000"
  volumes:
    - ./node.py:/app/node.py
    - ./requirements.txt:/app/requirements.txt
  command: sh -c "pip install -q -r /app/requirements.txt && python -u /app/node.py"
  depends_on:
    bootstrap:
      condition: service_healthy
  environment:
    - PORT=5050
```