

# Memgraph Proof of Concept Setup

---

Requires that Docker is already installed

## Preparation

---

1. Download the following files from the JIRA ticket <https://tracker.nci.nih.gov/browse/DHNEPTUNE-47> (<https://tracker.nci.nih.gov/browse/DHNEPTUNE-47>):
  - docker-compose-memgraph.yaml
  - snapshot
  - apoc\_replacement\_poc.py
2. Create a new directory and copy all of these files into it. This new directory will be referred to as the “root” directory in the following steps.
3. Create a new directory in the root directory called “snapshot”.
4. Copy the “snapshot” file into the new “snapshot” directory.
5. Create a new directory in the root directory called “internal\_modules”.
6. Copy the “apoc\_replacement\_poc.py” file into the new “internal\_modules” directory.

## Install the Memgraph Lab user interface

---

1. Navigate to the Memgraph downloads page <https://memgraph.com/download#individual> (<https://memgraph.com/download#individual>) and download “Memgraph Lab”
2. Run the installer.

Note: Memgraph Lab can be run as a Docker container with a browser interface but copy and paste functions do not always work correctly so the installed application is preferred.

## Create the Memgraph container

---

1. Start Docker if it is not already running.
2. Open a terminal and navigate to the root directory.

3. Run the following command:

```
docker compose -f "docker-compose-memgraph.yaml" up -d
```

4. Open Memgraph Lab
5. Memgraph Lab should automatically detect the Memgraph instance and display a “Connect now” button. Click the button to connect.
6. Verify that the snapshot data was loaded. If the snapshot file was loaded correctly then 48,560 Nodes and 88,137 Relationships should be displayed at the top of the Memgraph Lab window.

## Test the APOC Replacement Proof of Concept

1. Navigate to the “Query Execution” page in Memgraph Lab
2. In the “Cypher editor”, enter the command:

```
CALL mg.load_all();
```

and then click the “Run query” button. This will load the `apoc_replacement_poc.py` module.

3. Paste the below test query into the “Cypher editor” and then click the “Run query” button:

```
WITH ["d", ["a", "b", "c"], "e", ["f", "d"]] as input, {} AS output
WITH apoc_replacement_poc.merge(output, {
  flattened: apoc_replacement_poc.flatten(input)
}) AS output
WITH apoc_replacement_poc.merge(output, {
  sorted: apoc_replacement_poc.sort(output.flattened)
}) AS output
WITH apoc_replacement_poc.merge(output, {
  as_set: apoc_replacement_poc.to_set(output.sorted)
}) AS output
return output
```

4. The expected result is:

```
{
  "as_set": [
    "b",
    "c",
    "e",
    "f",
    "a",
    "d"
  ],
  "flattened": [
    "d",
    "a",
    "b",
    "c",
    "e",
    "f",
    "d"
  ],
  "sorted": [
    "a",
    "b",
    "c",
    "d",
    "d",
    "e",
    "f"
  ]
}
```

## Functions in the Proof of Concept Module

### **merge**

Merges two maps into a single map

### **flatten**

Converts a nested list into a single level list

### **sort**

Sorts a list

### **to\_set**

Removes duplicate elements of a list

