# **Neo4j graph database installation on Azure cloud Ubuntu Virtual Machine and configuration of Neosemantics package for Ontology**

**Xiaochen Zheng**

ICT4SM, EPFL, 1015 Lausanne, Switzerland

(November 3rd, 2021)

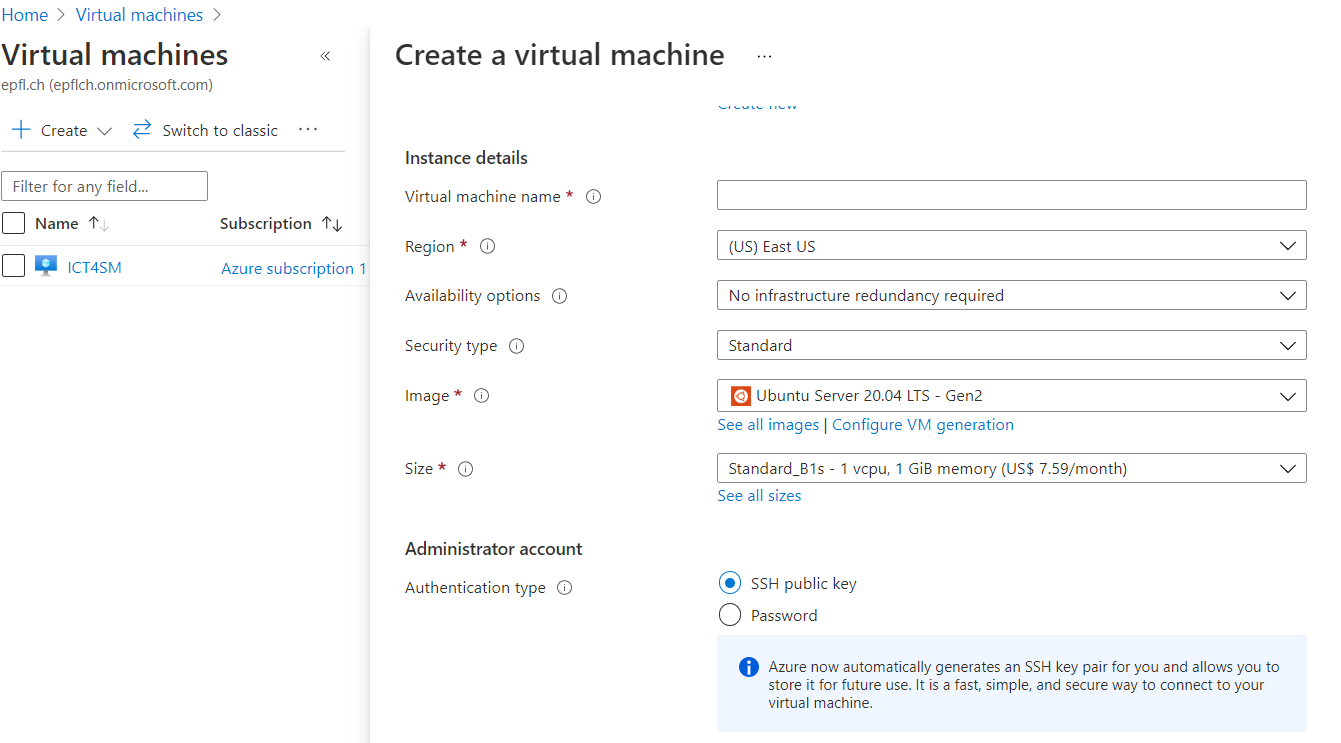
**Abstract:** This document explains how to install and configure a Neo4j graph database on an Azure cloud Ubuntu Virtual Machine (VM). Target readers are beginners of Azure cloud and Neo4j database. Following this instruction, you can create a VM on Azure cloud running Ubuntu system and then install and configure a Neo4j database which can be accessed remotely using the public IP address of the VM.

**Keyword:** Neo4j, Azure, Ubuntu, Neosemantics, Ontology, Python

## **Create Ubuntu VM on Azure**

You need a Microsoft account to register an Azure account. Azure provides one year (750 hours) free service for B1s tier Linux VM, which is enough to hold a basic Neo4j database. Go to the [Azure free account webpage](https://azure.microsoft.com/en-us/free/) and log in (or register first) to your account.

After login to the Azure portal, you will see the option to create Virtual Machines among other Azure services. Just use the default setting of the guide as shown below:



Once the VM is created, you can connect to it via three different ways: SSH, RDP and Bastion. I use SSH as example. You can find detailed instructions in this article [*“How to use SSH keys with Windows on Azure”*](https://docs.microsoft.com/en-us/azure/virtual-machines/linux/ssh-from-windows). To make the story short, all you need is two line of codes:

1. If you don’t have an SSH key pair on your local machine yet, create one with the following code in the command terminal or PowerShell of Windows:

ssh-keygen -m PEM -t rsa -b 4096

1. Connect to VM using your username and public IP address of the VM. You need to provide your password:

ssh -i ~/.ssh/id\_rsa xiaochen@13.95.70.195

## **Install Neo4j on VM Ubuntu**

Once you are connected to the VM via SSH, you can install Neo4j. A detailed instruction is provided in this tutorial: [*“How To Install and Configure Neo4j on Ubuntu 20.04”.*](https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-neo4j-on-ubuntu-20-04)The short version is as follows:

1. Install Neo4j:

sudo apt update

sudo apt install apt-transport-https ca-certificates curl software-properties-common

curl -fsSL https://debian.neo4j.com/neotechnology.gpg.key | sudo apt-key add -

sudo add-apt-repository "deb https://debian.neo4j.com stable 4.1"

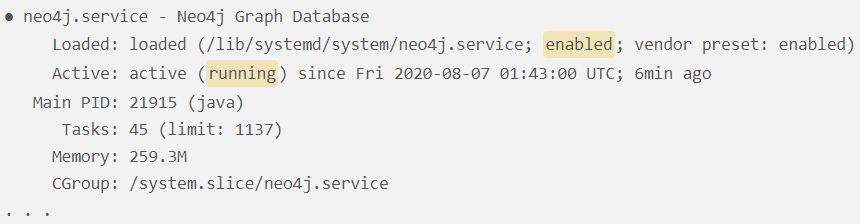
sudo apt install neo4j

1. Configure Neo4j to start automatically after reboot of the system:

sudo systemctl enable neo4j.service

sudo systemctl status neo4j.service

You should see output as below:



1. Configuring Neo4j for Remote Access:

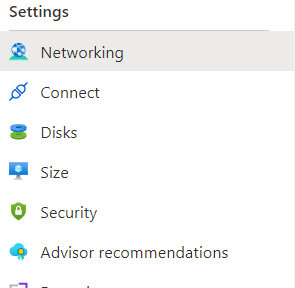
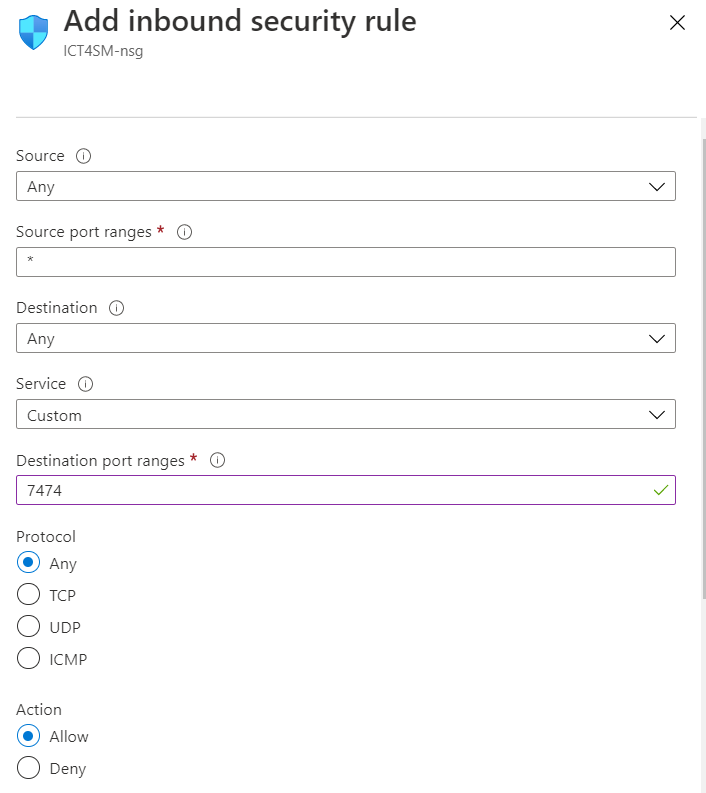
By default, Neo4j is configured to accept connections from localhost only (127.0.0.1 is the IP address for localhost). This configuration ensures that your Neo4j server is not exposed to the public Internet, and that only users with access to the local system can interact with Neo4j.

To change the network socket that Neo4j uses from localhost to one that other systems can use, you will need to edit the /etc/neo4j/neo4j.conf file. Open the configuration file in your preferred editor and find the dbms.default\_listen\_address setting. The following example uses nano to edit the file:

sudo nano /etc/neo4j/neo4j.conf

Locate the commented out #dbms.default\_listen\_address=0.0.0.0 line and uncomment it by removing the leading # comment character.

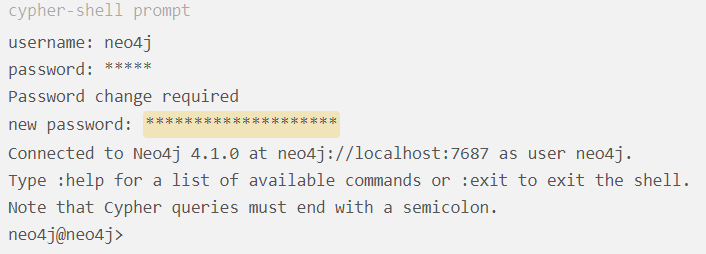
1. **Attention!!** After the above step, one more configuration is necessary to allow port 7474 and/or 7687 receive remote access. This can be done in the VM *Settings-Networking-Add inbound port rule*.

Now the Neo4j database should be ready for use.

If you are connected to the VM via SSH, you can simply access Neo4j by: cypher-shell.

When you first invoke the shell, you will login using the default administrative *neo4j* user and *neo4j* password combination. Once you are authenticated, Neo4j will prompt you to change the administrator password.



## **Install Neosemantics package for importing RDF ontology**

*Neosemantics is a plugin that enables the use of RDF in Neo4j. RDF is a W3C standard model for data interchange. This effectively means that Neosemantics makes it possible to*

* *Store RDF data in Neo4j in a lossless manner (imported RDF can subsequently be exported without losing a single triple in the process).*
* *On-demand export property graph data from Neo4j as RDF.*

To install Neosemantics in the cloud Neo4j database, a detailed tutorial is available [here](https://neo4j.com/labs/neosemantics/4.0/install/).

1. first download the package file from [Github](https://github.com/neo4j-labs/neosemantics/releases).

wget https://github.com/neo4j-labs/neosemantics/releases/download/4.3.0.1/neosemantics-4.3.0.1.jar

1. Then copy this file to plugins folder in Neo4j HOME directory:

sudo cp neosemantics-4.3.0.1.jar /var/lib/neo4j/plugins/

1. Add the following line to your <NEO\_HOME>/conf/neo4j.conf,

*dbms.unmanaged\_extension\_classes=n10s.endpoint=/rdf*

sudo nano /etc/neo4j/neo4j.conf

Ctrl+x to save&quite

1. Restart Neo4j server *sudo systemctl {restart|start|stop} neo4j*

sudo systemctl restart neo4j

Neosemantics should be functional by now. You can import an existing ontology to the Neo4j database:

1. Connect to Azure VM: ssh -i ~/.ssh/id\_rsa USER@IP.OF.THE.VM
2. Enter Neo4j: cypher-shell
3. Database initiation configure:

CALL n10s.graphconfig.init();

CREATE CONSTRAINT n10s\_unique\_uri ON (r:Resource)

ASSERT r.uri IS UNIQUE;

1. Import Ontology from github:

CALL n10s.onto.import.fetch("https://raw.githubxxxx/xxx/xxx\_ontology.ttl","Turtle");

Attention!!! The path to the ontology should be to the **raw** ontology (e.g. .ttl) file, not the default HTML version.

Now you can play with the imported ontology in the Neo4j database remotely. You can do all the query options in the cypher-shell environment just like locally.

## **Query Neo4j using Python script**

Neo4j provides [Neo4j Python Driver](https://neo4j.com/developer/python/) to enable query from Python. It can be installed by: pip install neo4j

Below is an example adapted from the official tutorial:

|  |
| --- |
| from neo4j import GraphDatabase  class Neo4jConnection:  def \_\_init\_\_(self, uri, user, pwd):  self.\_\_uri = uri  self.\_\_user = user  self.\_\_pwd = pwd  self.\_\_driver = None  try:  self.\_\_driver = GraphDatabase.driver(self.\_\_uri, auth=(self.\_\_user, self.\_\_pwd))  except Exception as e:  print("Failed to create the driver:", e)    def close(self):  if self.\_\_driver is not None:  self.\_\_driver.close()    def query(self, query, db=None):  assert self.\_\_driver is not None, "Driver not initialized!"  session = None  response = None  try:  session = self.\_\_driver.session(database=db) if db is not None else self.\_\_driver.session()  response = list(session.run(query))  except Exception as e:  print("Query failed:", e)  finally:  if session is not None:  session.close()  return response  if \_\_name\_\_ == "\_\_main\_\_":  ##replace the VM IP address and Neo4j user&password ####  conn = Neo4jConnection(uri="bolt://ip.add.of.vm:7687", user="neo4j", pwd="neo4j")  ##define your own query string  query\_string = '''MATCH (p:n4sch\_\_Class {n4sch\_\_label: xxxx}) RETURN p'''  result = conn.query(query\_string, db='neo4j')  print(result) |

## References:

1. How to use SSH keys with Windows on Azure, <https://docs.microsoft.com/en-us/azure/virtual-machines/linux/ssh-from-windows>
2. How To Install and Configure Neo4j on Ubuntu 20.04, <https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-neo4j-on-ubuntu-20-04>
3. Neosemantics, <https://neo4j.com/labs/neosemantics/4.0/install/>
4. Using Neo4j from Python, <https://neo4j.com/developer/python/>