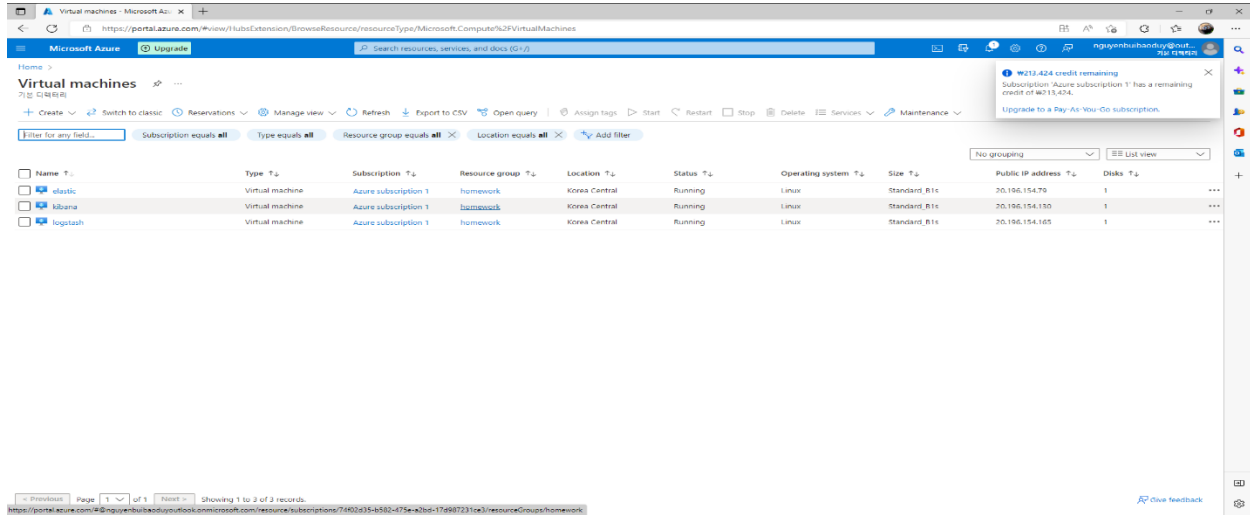


SW education

Homework Report

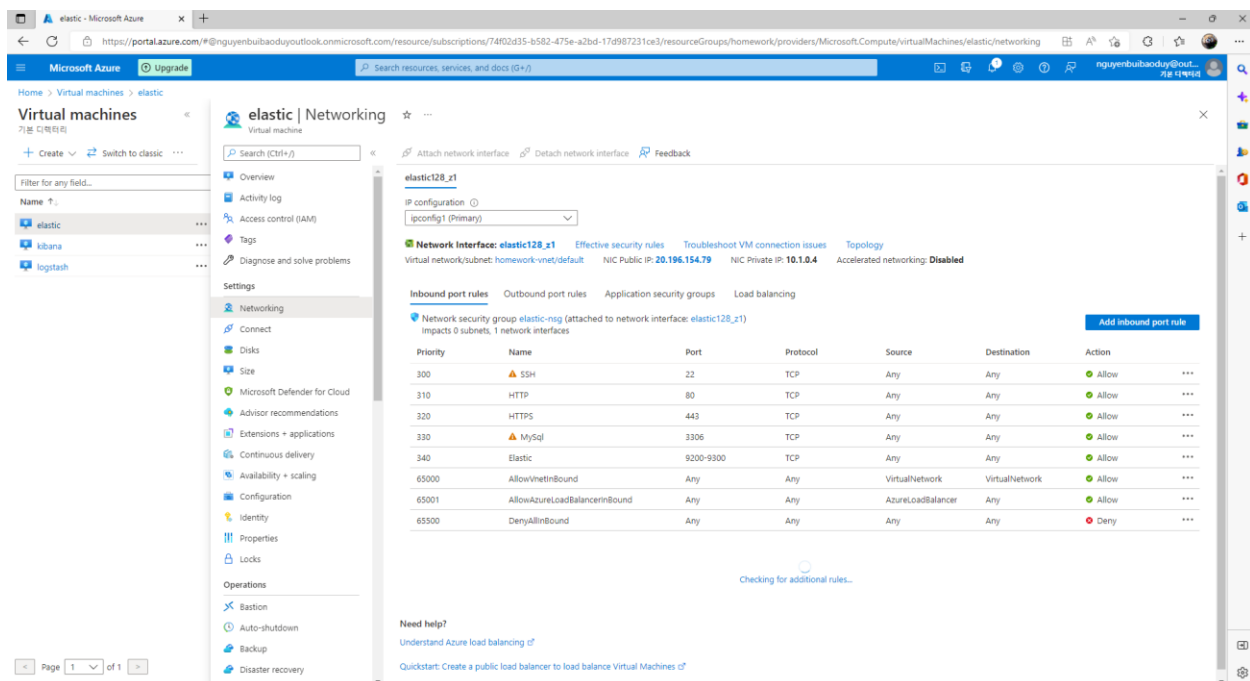
Notice: The data using in this report come from the data.go.kr which is called “동물보호관리시스템 - 유기동물 조회 서비스”

1. Create 3 virtual machine server for elastic, Logstash and Kibana:



2. Create networking for 3 servers:

a. Elasticsearch:



b. Logstash:

The screenshot shows the Azure portal interface for the 'logstash' virtual machine. The left sidebar contains navigation options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Networking, Connect, Disks, Size, Microsoft Defender for Cloud, Advisor recommendations, Extensions + applications, Continuous delivery, Availability + scaling, Configuration, Identity, Properties, Locks, Operations, Bastion, Auto-shutdown, Backup, and Disaster recovery. The main content area is titled 'logstash | Networking' and shows the 'logstash123_z1' network interface. It includes IP configuration (ipconfig1 Primary) and a table of inbound port rules.

Priority	Name	Port	Protocol	Source	Destination	Action
300	SSH	22	TCP	Any	Any	Allow
310	HTTP	80	TCP	Any	Any	Allow
320	HTTPS	443	TCP	Any	Any	Allow
330	MySQL	3306	TCP	Any	Any	Allow
340	Logstash	5044	Any	Any	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

c. Kibana:

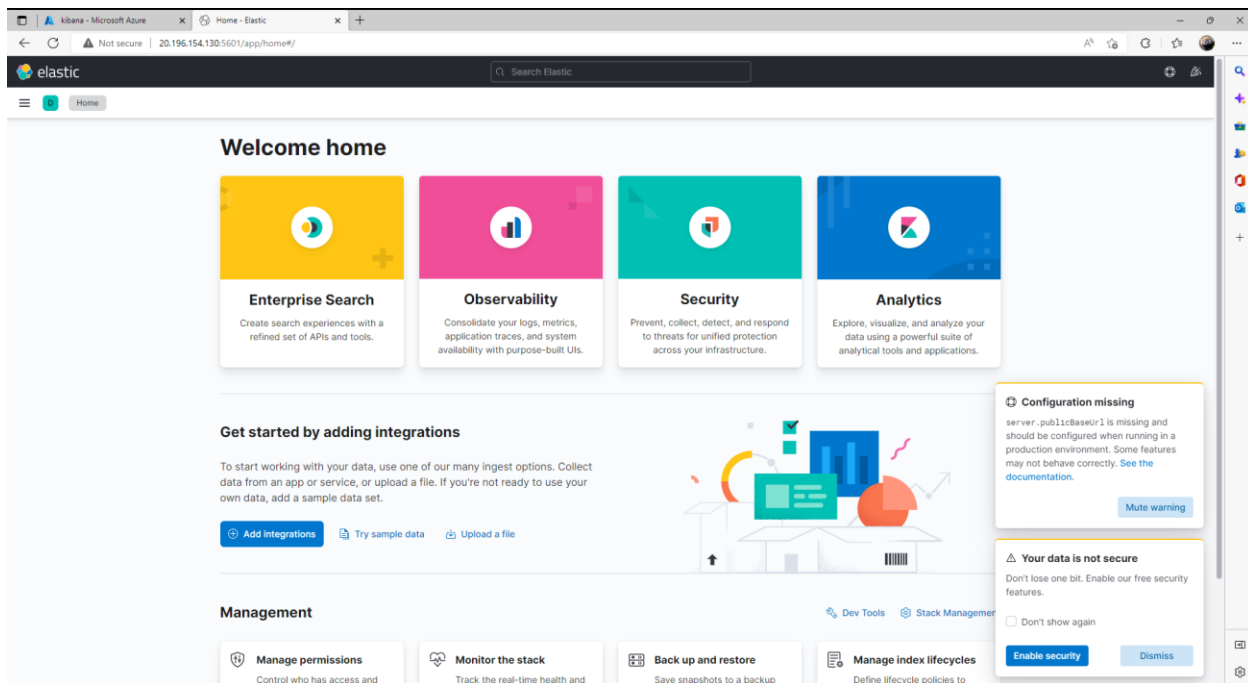
The screenshot shows the Azure portal interface for the 'kibana' virtual machine. The left sidebar contains navigation options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Networking, Connect, Disks, Size, Microsoft Defender for Cloud, Advisor recommendations, Extensions + applications, Continuous delivery, Availability + scaling, Configuration, Identity, Properties, Locks, Operations, Bastion, Auto-shutdown, Backup, and Disaster recovery. The main content area is titled 'kibana | Networking' and shows the 'kibana670_z1' network interface. It includes IP configuration (ipconfig1 Primary) and a table of inbound port rules.

Priority	Name	Port	Protocol	Source	Destination	Action
300	SSH	22	TCP	Any	Any	Allow
310	HTTP	80	TCP	Any	Any	Allow
320	HTTPS	443	TCP	Any	Any	Allow
330	MySQL	3306	TCP	Any	Any	Allow
340	Kibana	5601	Any	Any	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

3. Running server elastic:

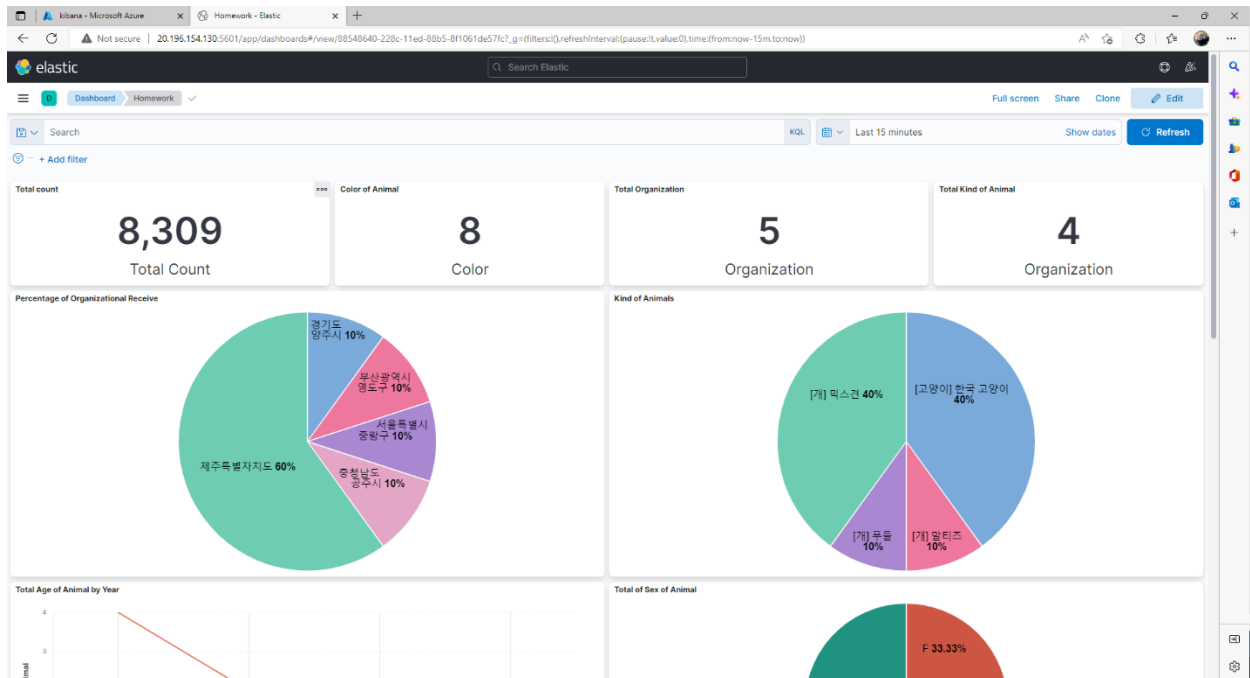
```
{
  "name": "elastic",
  "cluster_name": "elasticsearch",
  "cluster_uuid": "Kyr-AVvBTISPMESVlaKag",
  "version": {
    "number": "7.17.5",
    "build_flavor": "default",
    "build_type": "deb",
    "build_hash": "8d61b477d9f931f219e3745f295ed2bc50cde84",
    "build_date": "2022-08-23T21:57:28.736748635Z",
    "build_snapshot": false,
    "lucene_version": "8.11.1",
    "minimum_wire_compatibility_version": "6.8.0",
    "minimum_index_compatibility_version": "6.0.0-beta1"
  },
  "tagline": "You Know, for Search"
}
```

4. Running server Kibana:



5. Create dashboard on server Kibana: using data download from data.go.kr

Link: <http://20.196.154.130:5601/goto/4a48ac70-2361-11ed-88b5-8f1061de57fc>



6. Create a database based on data download on data.go.kr: using MySQL to create the same structure which data from data.go.kr but in my own way

MySQL Workbench

SCHEMAS

- 1. create database hds;
- 2. use hds;
- 3. create table animal(
- 4. 31
- 32 * show tables;
- 33
- 34 * insert into animal(id, Dt, Place, Kind, Color, Age, Weight, Notice, Sdt, Edit, Popfile, ProcessState, Sex, Neuter, SpecialMark, CareItem, CareTel, CareAddr, Orgin, ChargeItem, Official, NumOfRows, PageNo, TotalCount
- 63
- 64 * select * from animal;
- 65
- 66 * update animal set NumOfRows = 1000, PageNo = 1, TotalCount = 8000 where id = "1";
- 67

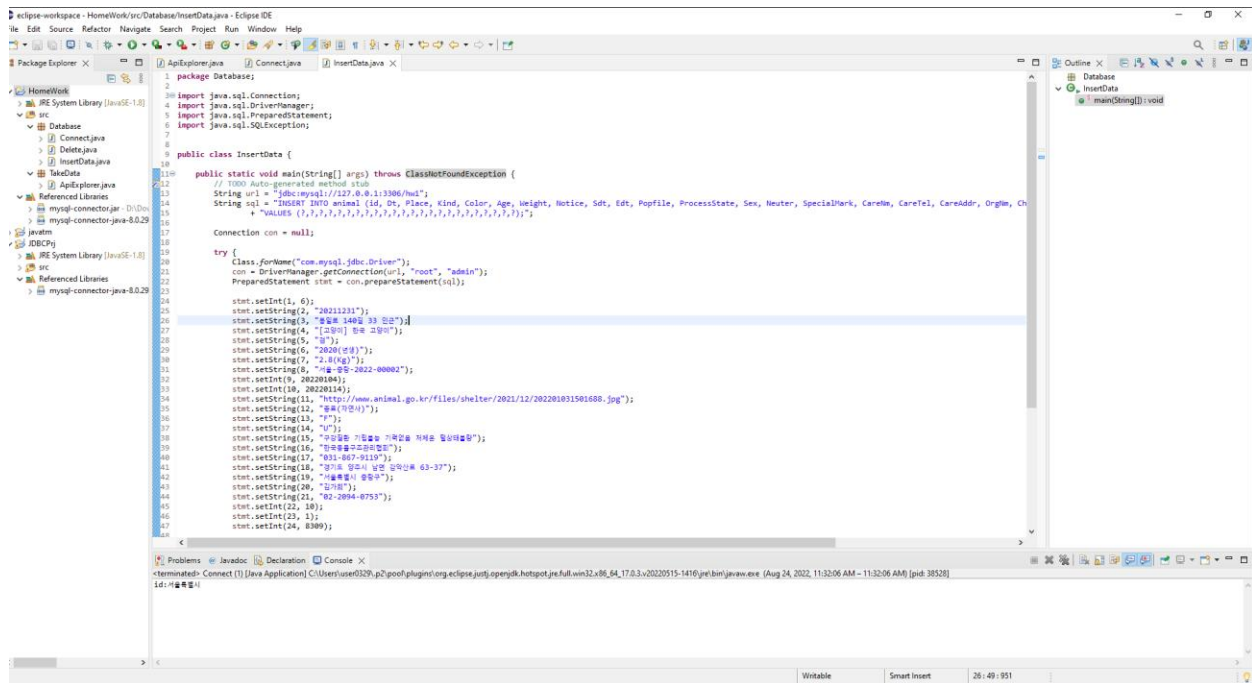
Table: animal

id	Dt	Place	Kind	Color	Age	Weight	Notice	Sdt	Edit	Popfile	ProcessState	Sex	Neuter	SpecialMark	CareItem	CareTel	CareAddr	Orgin	ChargeItem	Official	NumOfRows	PageNo	TotalCount
1	20220823	서울특별시	개	흰색	2022	3 kg		20220104	20220117	http://www.animal.go.kr/files/shelter/2022/12/...	종료(기)	M	N								1000	1	8000
2	20220719	서울특별시	고양이	흰	2022	2 kg		20220104	20220114	http://www.animal.go.kr/files/shelter/2022/12/...	종료(기)	F	N								1000	1	8000
3	20221231	경기도	개	검은	2022	3 kg		20220104	20220114	http://www.animal.go.kr/files/shelter/2022/12/...	종료(기)	M	N								1000	1	8000
4	20221231	충청남도	개	검은	2022	3 kg		20220104	20220114	http://www.animal.go.kr/files/shelter/2022/12/...	종료(기)	M	N								1000	1	8000
5	20221231	경상남도	개	검은	2022	3 kg		20220104	20220114	http://www.animal.go.kr/files/shelter/2022/12/...	종료(기)	M	N								1000	1	8000
6	20221231	경상남도	개	검은	2022	3 kg		20220104	20220114	http://www.animal.go.kr/files/shelter/2022/12/...	종료(기)	M	N								1000	1	8000

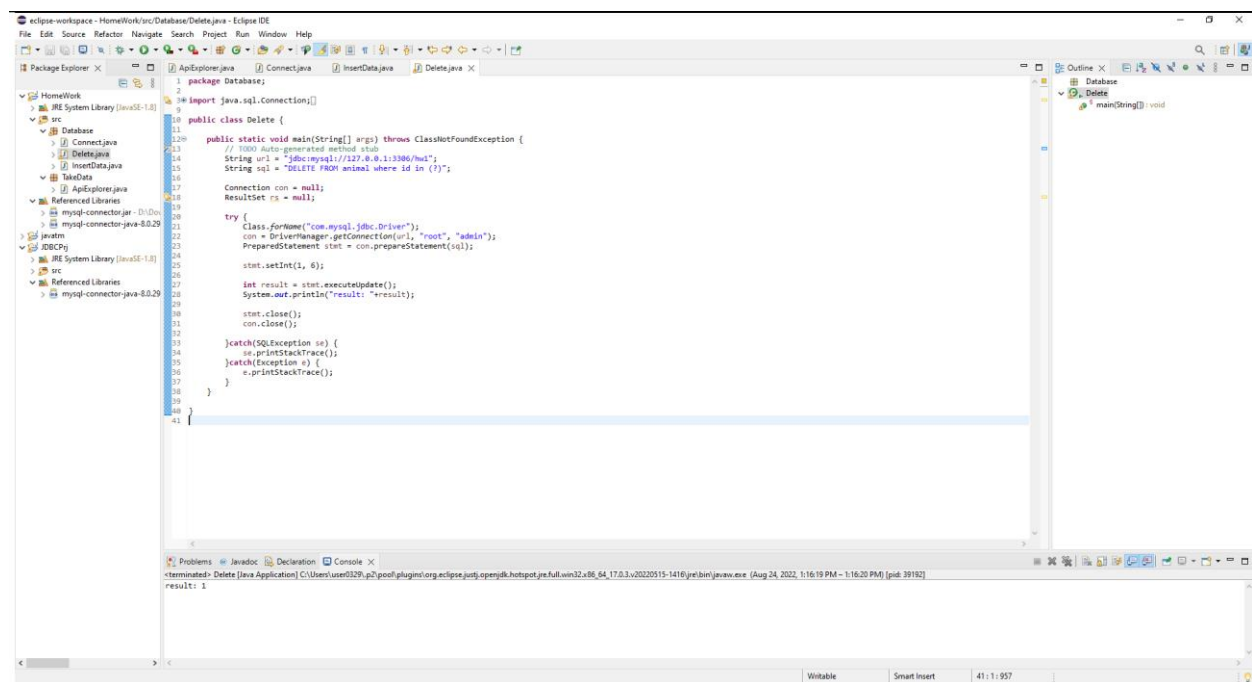
Output

Time	Action	Message	Duration / Fetch
10:33:04	use hds;	0 rows affected	0.000 sec
10:33:10	select * from animal LIMIT 0, 1000	5 rows returned	0.000 sec / 0.000 sec
10:53:52	select * from animal LIMIT 0, 1000	6 rows returned	0.000 sec / 0.000 sec

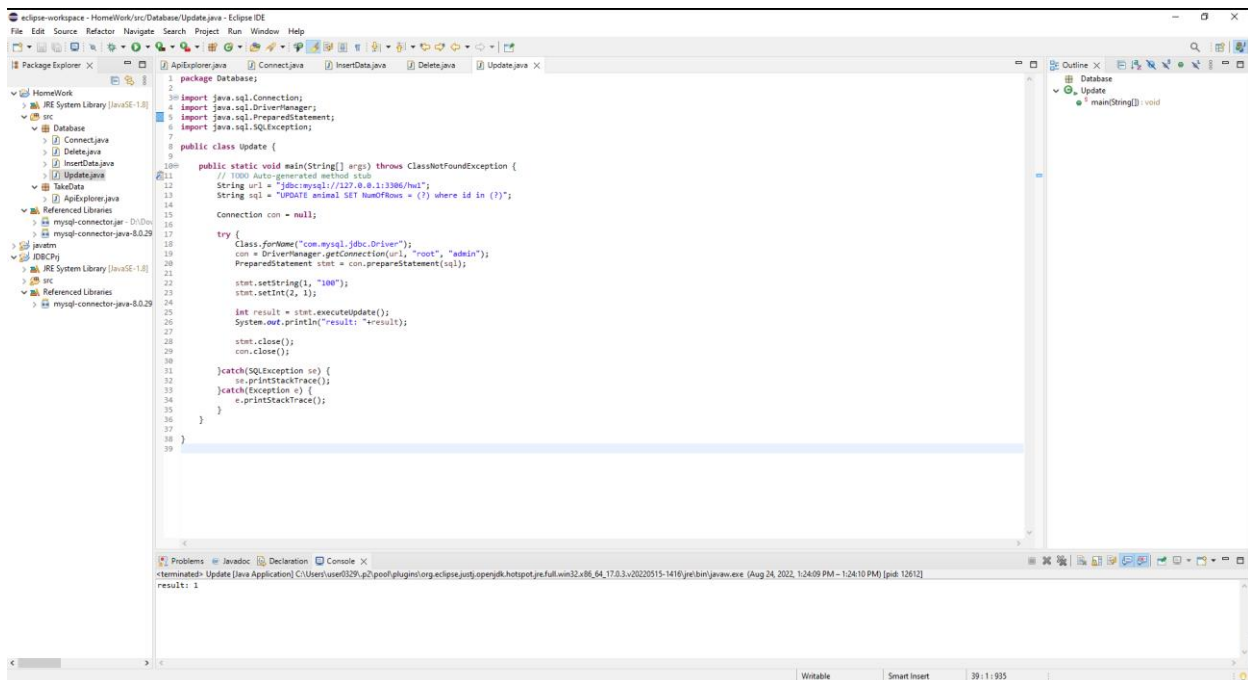
7. Connect download database from data.go.kr:



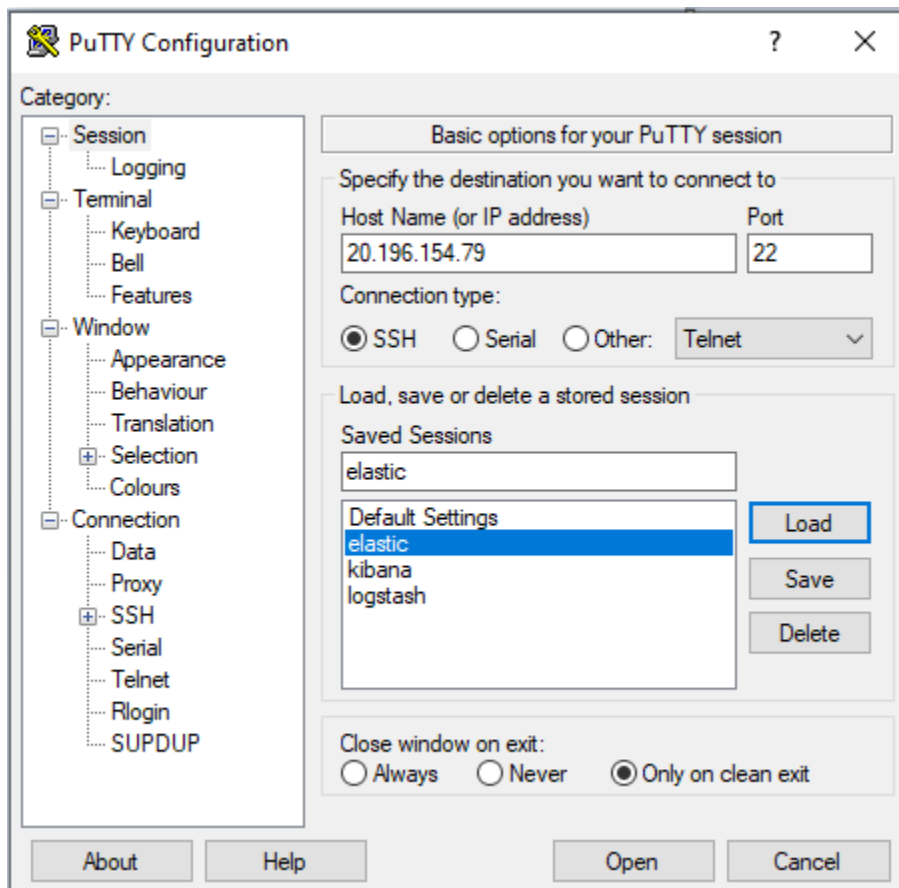
- Delete:



- Update:



10. Connect to azure server through putty:



Follow the book exercise with air_mapping.json file:

