Practical Session 1: Working with Cloud Infrastructure (ETAIS)

The aim of the practical session is to learn the basics of cloud computing that serve as fundamental infrastructure to deploy your application services in a scalable manner. Here, you will learn how to provision virtual machines from the cloud infrastructure and access them publically by managing the security groups. In this experiment, we will use a cloud infrastructure service known as Estonian Scientific Computing Infrastructure (ETAIS) managed by the Estonian Scientific Community (https://etais.ee/about/). This cloud infrastructure is configured on large-scale hardware resources using the OpenStack cloud operating system.

The ETAIS project is being carried out by a consortium of four institutions: the University of Tartu, Tallinn University of Technology, National Institute of Chemical Physics and Biophysics, and The Education and Youth Board (Harno).

How to access ETAIS cloud infrastructure from the University of Tartu network:

To access the local university cloud resources your computer has to be inside the UT network. So you should either use Eduroam Wifi (inside the institute building) or set up a VPN connection to the university network.

- Configuring UT VPN - https://wiki.ut.ee/pages/viewpage.action?pageId=17105590
- Eduroam Wifi - https://wiki.ut.ee/display/AA/Eduroam

Practical session communication!

There will be both physical and online lab sessions, but it is encouraged to attend in physical mode.

- Lab supervisors will provide support through Zulip, if you are attending in online mode.
- Login to Zulip using your university account (https://zulip.cs.ut.ee)
- We have already set up a Zulip topic for each lab and course-related discussions.
    - Use corresponding topic (e.g. `#Lab01-etais` for first practice session, `#Lab02-docker` for second practice session and so on) for lab-related questions and discussion.
- When asking questions or support from lab assistants, please make sure to also provide all needed information, including screenshots, configurations, and even code if needed.
    - If code needs to be shown to the lab supervisor, send it (or a link to it) through Direct Messages.

Pre-requisites:

- Able to login and see the project in ETAIS.
- Able to login to https://gitlab.cs.ut.ee with university login credentials
- Have access to “all-practice-sessions-Pub” Gitlab group.
- You are connected to the University network.

In case of issues check:

1. Possible solutions to common issues section at the end of the guide.
2. Ask in the `#Lab01-etais` topic.

Part 1. Introduction to ETAIS cloud infrastructure.

Estonian Scientific Computing Infrastructure (ETAIS) belongs to the Estonian roadmap of research infrastructures providing computing and storage resources for the Estonian Scientific Community. ETAIS aims to increase the competitiveness of the Estonian computing and data-intensive research disciplines by providing access to a new and modern scientific computing infrastructure. The OpenStack cloud operating system is deployed on a large set of hardware infrastructure located at universities including the University of Tartu.

For more information you can refer to www.etais.ee We have added a presentation video on a brief introduction to ETAIS and how to log in and provision the resources using a web interface.

1. Introductory video presentation on ETAIS
2. Creating a Virtual Machine in ETAIS

Notes on above recorded video 1:

- Timestamp - Now we are not using ~~Slack~~, we have shifted to Zulip
- The project name should look like “stdid1234” or like your study ID code.
- In your case, you will not find “DevOps2021Fall” project
- Signing in method in ETAIS have changed- You will see “Sign in with MyAccessID”

Part 2. Working with ETAIS cloud infrastructure

In these exercises, you will learn to access the ETAIS web interface and provision virtual machines using the ETAIS web interface.

Exercise 2.1. Accessing the ETAIS cloud services

The following steps will guide you to connect to the ETAIS web interface.

- Log into https://minu.etais.ee and click on Sign in with MyAccessID.
- In the search box, type, *University of Tartu* and login using your university username and password.
    - `Contact the lab supervisor if you have issues while logging in.`
- Familiarise yourself with the available ETAIS web functionality as mentioned in Part 1.
- Select “**UT-DevOpsCourse**” organisation and the project available to you (`If you have a problem in accessing the workspace then please report to teaching assistants in the Zulip stream!`). You should be able to access only one project, specifically created for you. The project name looks like “stdid1234”or like your study ID code.
- Create an ssh Key Pair for accessing Virtual Machines over the network. `Make sure the name of the Key Pair includes your last name!`

- Firstly, create keys(SSH with RSA 4096 bits) in your machine/laptop using the following tools. You can assign the last_name of yours as a private key name.
  `(Don't assign the password while creating the SSH keys!!)`:
  - Linux: `ssh-keygen` Windows: `Puttygen` (Store the private key with extension `.ppk`)
  - Mac OS: `ssh-keygen`
- Now, in the ETAIS web interface, on the right side click on `Your Name-->Credentials` --> `SSH keys` --> `Add Key` and complete the steps. (Make sure to keep key name same as a private key)

Exercise 2.2. Requesting computing resources from the cloud

In this exercise, you will start with a Cloud instance (or virtual machine) by specifying its configuration and computing resources available for it. In your project, you will find one VPC (Virtual Private Cloud) with 4 vCPUs, 8GB RAM, 20GB of storage (later, possibly in Lab03, this config will be increased to 14 vCPUs, 26GB RAM, 60GB storage).

- Click on the "Resources" tab and then "VMs" in the left side panel and go to "Add Resource" and select "Virtual Machine in <your project name> " and click on Deploy. Add the following values to the offering configuration page.
  - VM Name --> The VM name can be anything. But for this session let's name it "controller".
  - Click on "Show choices" in images and select **ubuntu22.04** virtual machine image.
  - Click on "Show choices" in Flavour and select **m3.tiny** (2 vCPUs + 2GB RAM).
  - System volume size --> **20 GB**
  - System Volume type --> **prod2 (HPC production HDD)**, leave other details as default.
  - SSH Public key--> select your recently created key.
  - Security Groups --> **default**, **allow-all**.
  - Networks --> leave it by default, Next select **Auto-assign floating IP** (to assign an IP to access machine in UT network)
- Finally, click on "`Add to cart`".
- It will take you to the purchase page and now click on "`purchase`".
- After a few seconds, click on "Refresh" and you should see the state-->`done` condition.

Exercise 2.3. Accessing your Cloud instance using ssh

We will use the Secure Shell (ssh) protocol to log into the started instance over the network. Instances in the cloud can have multiple IP addresses. **External IP** for accessing the instance from outside the cloud and **Internal IP** for accessing the instance from inside the cloud (from other instances). However, our instances have two IP's in the current configuration and note down "External IP" for connecting to the VM.

`You have to be in the university network to access the VMs. Use VPN or log in to eduroam, if you're using your own laptop`

- Log into the instance through **ssh** using SSH Key-based authentication
  - On Linux:
    - `ssh -i path_to_my_key_pair_file ubuntu@<instance public ip address>`
    - For example: `ssh -i .cloud/shiva_key.pem ubuntu@172.17.64.63`
    - if you get an error, check that the path to the keyfile is correct and that it has correct rights (`chmod 400 <filename>`)

- On Windows:
  - Use `Git BASH` command line, Putty, WinSCP, or SSH secure shell program to get a command-line interface to a remote server through ssh.
  - When using Git BASH, the command is exactly the same as in the Linux command line:
    - `ssh -i path_to_my_key_pair_file ubuntu@<instance public IP address>`
  - When using `Putty`:
    - On the Putty startup screen enter your `External IP`.
    - Click on the `Connection` menu, then choose `Data`.
      - Enter the default login of the user: `ubuntu`
    - Choose `Connection` -> `SSH` -> `AUTH` and browse for the `private key` you generated earlier.
    - Click `Open`.

`If you are unable to access the machine over ssh then modify and enable port number (ssh) in security group section`

Exercise 2.4. Deploying an application

In this exercise, you're going to learn how to deploy an application in this virtual machine. The sample web application is to read Tartu City weather data from PIs weather database server and display it in the form of a table on the HTML page. We will use the python flask framework to create this web application.

Information on PIs weather database server:

Weather data is provided by several service providers in the world such as OpenWeatherMap.com, Open-Meteo.com and others. In this course, we use the weather data of *Tartu city* retrieved from OpenWeatherMap API. However, in this practice session, PI has configured a weather database server of Tartu city, where weather data (with fields timestamp, feels_like, humidity, pressure, temp, temp_max, temp_min*)* is retrieved from OpenWeatherMap API on every 10 minutes interval and the data is stored in Influxdb time series database server. The following details of Influxdb are required for you in this experiment and are shared in the following steps.

- `Influxdb Server URL (INFLUX_HOST_ADD)`
- `Organisation name (INFLUX_ORG)`
- `Influxdb access token (INFLUX_TOKEN)`

Instructions to deploy the application:

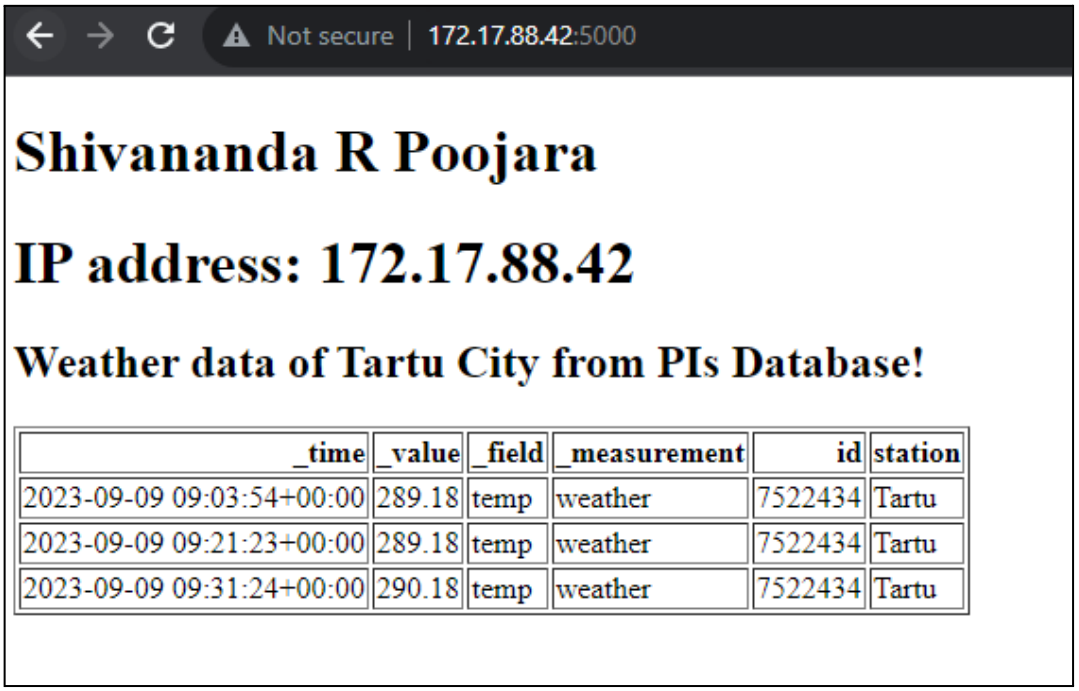
- Update apt using `sudo apt-get update -y`
- Install python virtual environment `sudo apt install python3-venv`
- Get the flask project with required code from here. You can also use git clone command - `git clone https://gitlab.cs.ut.ee/devops2023-fall/all-practice-sessions/Prac01-Etais/Flask-webapp.git`
- Move to app directory
  `cd Flask-webapp/flask-app`
- Create virtual environment `python3 -m venv venv`
- Activate virtual environment `source venv/bin/activate`
- Install flask framework `pip install Flask`

- Install pandas framework `pip install pandas`
- Install influxdb client `pip install influxdb-client`
- Let us look into the main python code (`app.py`).This code queries the data from PIs *influxdb* database and provides records to html for displaying. Here are quick highlights of the code.
    - Basic Flask, Influxdb imports
    - Read environment variables from host
    - Create Influxdb client to connect to database server
    - Query the data for the previous 60 minutes and put the result into a dataframe.
    - Render html template, with the data points.
- Add your name to the HTML page (html page is inside *templates* directory)
- Update the HTML page to display the IP address of the VM.
    - Hint: How to get the IP address?
    - No hard coded IP address is allowed. You need to programmatically get the IP of the VM.
- Set the environment variables (`INFLUX_HOST_ADD`,`INFLUX_ORG`,`INFLUX_TOKEN`).

```
INFLUX_HOST_ADD=http://172.17.88.218:8086
INFLUX_ORG=UT
INFLUX_TOKEN=SdwxQOFXhnBO8JNTr20CTkjPOq9E321mpcVoeu4NZQX3r6-Ym1Qe42PuYVI
SMDg15GHnC5cuv4pAXwlvnDW8hQ==
```

    - E.g. `export INFLUX_HOST_ADD=<INFLUX_HOST_ADD>`
- Now run the application `python3 app.py`. You should run this from the `flask-app` directory.
- Now open the application using http://VM external_IP:5000 in your machine browser

Example of Screenshot:



_time	_value	_field	_measurement	id	station
2023-09-09 09:03:54+00:00	289.18	temp	weather	7522434	Tartu
2023-09-09 09:21:23+00:00	289.18	temp	weather	7522434	Tartu
2023-09-09 09:31:24+00:00	290.18	temp	weather	7522434	Tartu

Deliverables

[UPDATE] : Please don't delete the virtual machine. It is recommended to just shut down the VM. Do not delete the *Flask-webapp* directory.

1- Take a screenshot of your webpage. Note!! **your name** and **IP address** should be visible

2- zip the flask-app directory(**excluding the virtual environment**) and the screenshot

(You can use the *scp* command to copy the files from VM to local machine).

The following command should run in your **laptop** on git bash, cmd prompt or terminal

Command: `scp -i ssh_key username@server_ip:/path_to_remote_directory local_machine/path_to_the_file`

Ex: `scp -i poojara.pem ubuntu@172.17.45.18:/home/ubuntu/Flask-web/flask-app .`

3- upload the zip file to the course wiki page.

4- ***[Note]*** Do not delete this zip file. You need this in other labs.

Possible solutions to common issues

1. If you can not access the Cloud instance over SSH,
    - it may be because you have not set up a VPN connection.
2. If you can not access your instance over port 5000
    - it may be because you have not opened port 5000 in your security group (add allow-all security rule)
    - or added the created the security group to your instance
    - or not used correct IP filter for Remote IP Prefix:
        - Remote IP Prefix should stay 0.0.0.0/0 (this means all devices from any IP address can access this port)
        - Do not assign a Remote Security Group, it would limit to the specified port from only the cloud instances inside the selected security group.