Infrastructure as a Service

This exercise will guide you through the basics of setting up a cloud account, configuring your first firewall and configuring your first instance. Since the project work is based on Exoscale this account will showcase using Exoscale. The principles shown here can be applied to any cloud provider.

For the purposes of following this exercise please use the voucher provided. Remember, the voucher is limited to 20 euros, so destroy any resources you create for testing.

Starting your first instance

Once you log in to the Exoscale interface you will find several services on the left side: Compute, Storage, DNS, etc. Compute is the name for the IaaS platform discussed in lecture 2.

When you look at the Compute interface you will find left hand menu with several options: Instances, Instance pools, Templates, Firewalling, IP addresses, and so on. As a first exercise we will launch a single instance, so please open the Instances option and click the "Add" button on the interface.

The "New instance" interface presents you with several options. Most importantly, you will select the operating system template, instance size and storage size. These options will be present on any cloud provider and affect the pricing. Instances on Exoscale, and on many other providers, are priced per-second for the instance size (Micro, Tiny, etc), and the storage is priced separately. When an instance is stopped, but not destroyed, only the storage price is charged *when the account is not in trial*. Since your account is in trial your instance is also charged-for when it is stopped.

For the purposes of this exercise please select "Linux Ubuntu 20.04 LTS 64-bit", Micro, and a 10 GB disk. You can start the instance in any zone, but we recommend starting it in Vienna for latency reasons.

The next option is the key pair. Key pairs are SSH keys used for passwordless authentication (more on that later). For now, please ignore the Keypair, Private Networks and IPv6 options.

The next option is important: Security Groups. For now we will stick with the `default` security group, but remember this option for later as it will become important to building your architecture. As discussed in lecture 2, security groups provide network firewalling.

You can also ignore Anti-Affinity groups for now. Remember for later, anti-affinity groups are useful when you are building a redundant service and you don't want two instances to run on the same physical hardware. That's what anti-affinity groups guarantee.

The last item in the list is `User Data`. Again, as discussed in in lecture 2, `User Data` is a way to pass an initialization script to the virtual machine. For our exercise let's put a very simple Bash script here:

```
#!/bin/bash

set -e

apt update
apt install -y nginx
```

Once you are done with the configuration, please click the "Create" button. The instance will now be installed and the initialization script will run.

Please note the "Password" field. Since you have not provided an SSH key the password is the only way to log in to your instance now. Wait until the instance has booted and make a note of the password!

While you wait for the installation script to run please go to the "Firewalling" option on the left side and click the "default" security group. Here you can create a firewall rule that lets HTTP traffic in. Please click the "New Rule" button in the top right hand corner and add a rule with the following options:

- Type: `Ingress`
- Protocol: `TCP`
- Source type: `CIDR`
- Source: `0.0.0.0/0`
- Port(s): `80` - `80`

This will allow web traffic in from the whole internet.

> **Networking know-how needed!**
>
> For the purposes of this course it is assumed you have a basic understanding of computer networks, including IP addresses and CIDR masks. If you feel like you need a refresher, please give the Geek University CCNA course a quick read or ask us in the tutoring sessions.

For later use please also add an SSH rule (port 22). Make sure that SSH is only allowed from your IP address. Opening SSH to the whole internet will incur a flood of hacking attempts that increase the log size needlessly. Alternatively, you can also move SSH to a different port.

In the mean time your instance should be up and running. Please go to the "Instances" option and copy the IP address of your server. Open a new browser tab and insert the IP address there. If you did everything correctly an nginx welcome page should show up.

If you now wish to log in to the server you can use the command prompt / console depending on your operating system. Please copy the command from the "SSH command" field in the interface into your command prompt. When you are prompted the password, insert the password. (Unlike on the graphical interfaces, an SSH login won't show you the stars for the characters you typed.)

Setting up the CLI

Using the graphical interface is not the preferred way to interact with the cloud. In case of more complex cloud providers (Azure, etc) the user interface may not contain all features, or may be too complex to use. Command line interfaces provider a developer-friendly way to interact with the cloud provider.

In case of Exoscale you will have to install the exo cli on your computer. Once you have it installed, add it to the `PATH` of your machine and then open a command prompt / terminal.

In the terminal please type `exo config` to set up the configuration. During the setup you will be prompted for an API key which you can create in the Exoscale interface in the IAM menu.

Once your exo CLI is configured you can create the firewall rules we created previously using the following commands:

```
exo firewall add default --protocol tcp --port 80
exo firewall add default --protocol tcp --port 22
```

Provisioning an instance is similarly simple:

```
> exo vm create test -d 10 -o micro -t "Linux Ubuntu 20.04 LTS 64-bit"
Creating private SSH key
Deploying "test" ⁖                                          done

What to do now?

1. Connect to the machine

        exo ssh test

2. Put the following SSH configuration into ".ssh/config"

Host test
        HostName 194.182.174.84
        User ubuntu
        IdentityFile C:\Users\janos\AppData\Roaming\exoscale/instances/
34df135d-8ff1-47b5-81a9-194c71233941/id_rsa

Tip of the day:
        You're the sole owner of the private key. Be cautious with it.
```

"test" in this example is the name of the instance. The benefit of using the exo cli is the automatic use of an SSH key. You can now use the `exo ssh test` command to SSH into your instance securely with an SSH key instead of a password. If you want to add a User Data you will have to store the script in a file and then pass the `-f filename.sh` option.

> **Be careful!**
>
> If you store the user data script on a Windows machine it will do so with Windows line endings. This will not work on Linux systems. Make sure to use an editor that can save the script with Linux line endings.

Creating a virtual machine pool

Creating a virtual machine pool is very similar to creating an instance. Assuming you have saved your User Data script in a file called `userdata.sh` you can start a pool with 2 machines with the following command:

```
exo instancepool create autoscaling \
    --service-offering Micro \
    --template "Linux Ubuntu 20.04 LTS 64-bit" \
    --zone at-vie-1 \
    --security-group default \
    --cloud-init userdata.sh \
    --disk 10 \
    --size 2
```

Destroying your instances

Once you are done with this exercise make sure to destroy your instances to save on the budget.