more

Cloud Computing

Winter Term 2019/2020

Practical Assignment No. 1

Due: 28.11.2019 23:59

The primary goal of this assignment is to gain insight into performance characteristics of virtualization by benchmarking different cloud providers and comparing and evaluating the benchmark results. The secondary goal is to familiarize yourself with Amazon EC2 and Google Cloud, and the respective tools as an example of today's Infrastructure-as-a-Service clouds.

1. Prerequisites

1.1. Student groups

We assume you already formed groups of 3-4 students on the course website in ISIS (link in the “Practical Assignments” section).

1.2. Unix Platform

Work on a Unix-like platform to solve the assignments. Linux is recommended, Mac OS works as well. If you use Windows privately, work in a virtual machine (either locally, such as VirtualBox, or use one of the Cloud platforms).

1.3. Amazon AWS Educate

Create and Amazon AWS Educate Starter account in order to work with the cloud platform and perform further tasks.

- Go to https://www.awseducate.com/registration and go through the registration process. **Important:**
  - On Step 2, enter your TU Berlin student email address
  - On Step 3, select the bottom checkbox for a “AWS Educate Starter Account”
  - You will receive a confirmation email, follow the link in the email and finish the application process.
- The application process usually completes within a few hours, but might take longer in individual cases. Since only one account is required per group, all group members can attempt to register in order to speed up the process.
- AWS Documentation: Using the AWS CLI documentation and the Amazon EC2 online documentation you can find out how to setup and use the API and AMI tools.

- Configure your CLI tools so that they can access your AWS account, i.e., commands like the following should work:
    - **`aws ec2 describe-instances`**

<u>Note:</u> You are free to create a full Amazon AWS Educate account (not Starter). This will require a credit card number, which might be charged in the case that you use up the free credit that you receive per semester. In return, you will gain access to many additional features and resources that are quite interesting to play with! We recommend you try out these features, however in this case it will be ESPECIALLY IMPORTANT to **shut down your VMs** when you are done working with them, otherwise your credit card might be charged!

1.4. Google Cloud

- Visit this link to retrieve your GCP credit coupon:
    - Enter your TUB Email address
    - Each student can retrieve one coupon
    - Each coupon is worth 50$ and valid until 11.11.2020
- You will receive an email with further instructions on how to create an account and redeem your coupon
- You can optionally check if you are eligible for additional free trial credit at https://cloud.google.com/free/
- Study the documentation of the Google Cloud SDK and learn how to set up and use the command line tools
- Configure your Google Cloud SDK that it can access your GCP account, i.e. commands like the following should work:
    - **`gcloud compute instances list`**

<u>Note:</u> Although the AWS Educate Starter account is free and you have coupons for the Google Cloud Platform, you might run out of credits. Remember to **shut down your VMs when you do not use them!**

2. Virtual Machines

Write two well-commented Shell scripts that prepare and start a virtual machine on the AWS and GCP platforms, respectively.

- You can first manually experiment with the **aws** and **gcloud** commands and make notes, before finally combining the commands into a script.
- Each command in the scripts should be commented, make sure the scripts are clean and not cluttered with unnecessary text.
- After writing the scripts, test them: delete all running VMs, run the script, and check that the result is satisfactory.
- Do **not** include any private information such as your access key or secret key in your submission (if necessary, replace them with dummy strings).
- All requirements listed below should be covered by at least one command.

General requirements (apply for both platforms):

1. Generate a local SSH key pair in the current directory using the tool `ssh-keygen`. Name the output files `id_rsa` and `id_rsa.pub`. Make sure to specify a valid username as the comment in the public key (required for GCP).

Requirements for AWS:

1. Ensure that you have a default VPC in your AWS project.
2. Upload the generated public key as a key pair.
3. Create a security group and allow incoming ICMP and SSH traffic.
4. Run an instance with the following parameters:
   - The created key pair and security group
   - Instance type t2.micro
   - Image for “Ubuntu Server 18.04”. At the time of writing the AMI image id is `ami-0cc0a36f626a4fdf5`, which could change

Requirements for GCP:

1. Prepare a modified copy of the public key as described in the GCP documentation. Use shell commands like `echo`, `cat` and file redirection (“>”). Note: the username in the prepared key must match the username specified with `ssh-keygen`.
2. Upload the public key into your project metadata
3. Create a Firewall rule that allows incoming ICMP and SSH traffic. The rule must apply only for VMs with the tag “cloud-computing”.

4. Launch an instance with the following parameters:
    - Machine type g1-small
    - Add the tag “cloud-computing”
    - Image for “Ubuntu Server 18.04”. At the time of writing it is called `ubuntu-1804-bionic-v20191021`.

Hints:

- After the scripts work, ping the public IP of your instance and connect via SSH. This is not part of the submitted scripts, but should be done as validation.
- For the AWS VM, the SSH user name is always “ubuntu”, while for the GCP VM, you should use the username that you specified in the key.
- Shut down the virtual machines until you start working on the assignments below

Outputs:

- Scripts **create-aws.sh** and **create-gcp.sh**
    - Containing all commands for preparing and starting your VMs, including comments explaining what the commands do.

3. Prepare Performance Benchmarks

We will use the sysbench tool to run rudimentary benchmarks on our VMs. Manually install the tool on both VMs:

```
sudo apt update && sudo apt install -y sysbench
```

Write a benchmarking Shell script called benchmark.sh that uses the sysbench command to perform the 4 benchmarks described below.
The script must be executed without parameters and output a *single line* on the standard output that contains all measurements in CSV format, prefixed by the Unix epoch timestamp when the measurements were started:

```
1573143062,1058.52,9890.61,132.06,414.23
```

On the standard error output, the script can print arbitrary output.
The benchmarked resources are:

- CPU speed (events/s)
- Memory access (MiB/s)
  - Use block size of 4KB and a total size of 100TB (tera bytes)
- Sequential disk read speed (MiB/s)
- Random-access disk read speed (MiB/s)
  - For both disk tests, use 1 file with size 1GB, and make sure sysbench uses ***direct*** disk access to avoid caching effects.

Each benchmark must be executed for 60 seconds. The results of all 4 benchmarks must be collected in the script, then printed as one line, as shown above. Except for sysbench, the script is only allowed to use commands that are available in the default Ubuntu images used (don’t install any additional packages with apt or similar tools).
Make sure that every command in your script is commented.

Hints:

- The sysbench tool outputs a lot of information that we do not need. Use typical Shell tools such as **grep** and **awk** to obtain only the necessary piece of output from each sysbench invocation.

Outputs:

- Script **benchmark.sh**

4. Execute Performance Benchmarks

Execute the prepared benchmarks on the following two Cloud platforms. Each benchmark must be executed once every 30 minutes for two days. The results must be collected in one CSV per VM. In the end, each CSV file should have at least 2*48=96 entries.

Example of such a CSV file:

```
time,cpu,mem,diskRand,diskSeq
1573143062,1058.52,9890.61,132.06,414.23
1573143092,1057.51,9891.62,133.07,415.24
1573143122,1056.50,9892.63,134.08,416.25
...
```

Before or after starting the benchmarks, make sure to manually add the header line.

Use cron to automate your benchmarks. Use the `crontab` and echo commands to add a single entry to the Crontab of your user. The entry should execute the benchmark.sh script and append its output to the respective CSV file.

Additional tasks:

- Add the two full `crontab` commands to the end of your `create-*.sh` scripts.
- After taking all benchmarks, use the `plot.py` Python script provided on ISIS to generate plots for your benchmarks. If the scripts completes without errors, your files have the correct format.
- Answer the questions below in a well-formatted, readable text. Copy the text into the submission text field on ISIS.

Hints:

- When executing long-running experiments in a public cloud, remember to keep an eye on your credits. You should have more than enough credit to run these benchmarks, but make sure to shut down your VMs when you don’t actually use them.

Outputs:

- Two CSV files **`results-gcp.csv`** and **`results-aws.csv`**
- 4 plots generated by `plot.py`:
  - **`plot-[cpu|mem|diskRand|diskSeq].png`**
- Text with answers to questions
  - Paste the text into the submission text field on ISIS
  - Please use max. 200 words per question

CPU benchmark questions:

1. Shortly describe, how sysbench performs CPU benchmark. What does the resulting events/s value represent?
2. Look at the plots of your long-term measurements. Do you see any seasonal changes?

Memory benchmark questions:

1. Shortly describe, how sysbench measures memory performance.
2. How would you expect virtualization to affect the memory benchmark? Why?

Disk benchmark questions:

1. Shortly describe, how sysbench performs the disk benchmarks.
2. Compare the results for the two operations (sequential, random). What are reasons for the differences?

General question:

1. Compare the overall long-term measurement plots for the two platforms AWS and GCP. Name one type of application that you would expect to perform better on AWS, and one that would perform better on GCP, respectively. Shortly explain your decisions.

5. Submission Deliverables

Submit your solution on the ISIS platform as individual files. Please submit ONLY the necessary files, and us exactly the given file names!
Submit the text with answers to the questions directly in the submission text field.

Expected submission files:

- `create-aws.sh`
- `create-gcp.sh`
  - **2 script files, 10 points each**
- `benchmark.sh`
  - **1 script file, 10 points**
- `results-gcp.csv`
- `results-aws.csv`
  - **2 CSV files, 6 points each**
- `plot-[cpu|mem|diskRand|diskSeq].png`
  - **4 image files, 2 points each**
- Text with answers to questions
  - **2 points per question**

Total points: 64

Final Warning: Please make sure your VMs are shut down when you are not using them! Make good use of your credits :-)