

Cloud computing

Simple AWS cluster - configuration, creation, calculations

Michal Ranachowski

2019

February

1 Introduction

The purpose of this paper is to understand basic concept of cluster and go through process of creation cluster Amazon Web Services resources. It was created as part of cloud computing course. This paper is addition to bash script <https://github.com/mranachowski/AWS.git> and it describe conceptual layer of creating this type simple cluster. Script is commented inside so combined with this document person with basic knowledge of bash and Linux can easily use it. Calculation costs were covered from AWS Educate grant funds.

2 Amazon Web Services

Amazon Web Services (AWS) is on-demand cloud service platform providing computer power, database storage and specialist tools for data processing ¹. It works under AWS Global Infrastructure - data centers spread around the world.

*The following section was created based on AWS documentation.*²

2.1 AWS first steps

To begin using service user must create account and perform basic settings in AWS console. This part is not covered in this paper.

2.2 AWS CLI

CLI(*Command Line Interface*) is command-line tool for managing Amazon Web Services resources. Combined with bash it allows to create scripts for task automation. On Linux machines it can be easily install through Python Package Installer:

```
$ pip install awscli
```

After installation process user must configure AWS CLI, it can be done by execute command and provide necessary credentials and information. It will create default profile.

```
$ aws configure
```

```
AWS Access Key ID :
```

```
AWS Secret Access Key :
```

¹<https://aws.amazon.com/what-is-aws>

²<https://docs.aws.amazon.com>

Default region name :
Default output format :

The given data is stored in files: `./aws/credentials` and `./aws/config`.

2.3 Used services - EC2 and S3

EC2 - Elastic Compute Cloud provides virtual computing environments, *instances*, which could be treat for imagine purpose as physical machines. It works under Amazon Machine Images (AMI) - "operating system". User can choose from many different hardware configurations: CPU, memory, volume storage. Amazon provides few specialized instances group.³

S3 - Simple Storage Service provide object storage. From the user's side it is collection of *Buckets* that contain folders. As well as EC2, there are few types of storage classes.⁴

2.4 Useful concepts

2.4.1 Placement group

There are three type of placement group: *cluster*, *partition*, *spread*.⁵ Cluster ,used in script, logically place group of instances in a single Availability Zone providing high network throughput and low latency communication between instances which is crucial for cluster performance. Placement group creating by AWS CLI:

```
$ aws ec2 create-placement-group --group-name your group name --  
    ↪ strategy cluster
```

2.4.2 Security group

Security group acts like a virtual firewall and controls traffic for instances. When creating instances user must assign security group, instead of default would be apply. By default all inbound traffic is forbidden and all outbound traffic is allowed. By adding rules this settings can be modified. There are rules for communication inside and between security groups and rules for traffic from the Internet (ips, protocols and ports).

```
$ aws ec2 create-security-group [options]  
$ aws ec2 authorize-security-group-ingress [options]
```

2.4.3 IAM

IAM - *Identity & Access Management* is service that helps securely control to AWS resources. <https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html>

IAM users ale created within one account, and root user should no be use for everyday task, olny for administrative ones. IAM policies are collection of rules and permission. IAM role gives AWS service (for example EC2) permission to act like a user so we can now assign policies to this service. For example necessary when EC2 want to has access to S3 resources.

³<https://aws.amazon.com/ec2/instance-types/>

⁴<https://aws.amazon.com/s3/storage-classes/>

⁵<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html>

2.4.4 Instance skeleton

Instance skeleton are templates in java script format *.json* which holds many settings for launching instance. Default skeleton is generating by;

```
$ aws ec2 run-instances --generate-cli-skeleton
```

3 Parallel computing

Parallel computing is a type of computation in which many calculations or the execution of process are carried out simultaneously.⁶ Main idea is to split larger problems (size and/or time consuming) into smaller ones. Parallelism is basis of high-performance computing. There are many type of paralleling and it is large field of nowadays IT technology (hardware and software). In this paper only high-level use of MPI was covered.

3.1 MPI

For performing parallel computing on created AWS CLuster MPI was used. MPI stands for *Message Passing Interface* and it is communication protocol widely used which became standard for high-performance computing⁷. Originally developed for C, C++ and FORTRAN, but today they are available implementation for most popular programming language. MPICH2 implementation was used in this project. Under Ubuntu installation process:

```
sudo apt-get install mpich
```

Main parts of this packages are: *mpicc* - C MPI compiler and *mpiexec* - which allows executing previously compiled files. Below is an example usage of *MPI Hello world* (also given).

```
mpiexec -np NUMBER_OF_PROCESSES -f HOST_FILE ./hello_world\textit{(
    ↪ already compiled)}
```

```
#include <stdio.h>
```

```
#include <mpi.h>
```

```
int main(int argc, char** argv) {
    int myrank, nprocs;
```

```
    MPI_Init(&argc, &argv);
```

```
    MPI_Comm_size(MPI_COMM_WORLD, &nprocs);
```

```
    MPI_Comm_rank(MPI_COMM_WORLD, &myrank);
```

```
    printf("Hello from processor %d of %d\n", myrank, nprocs);
```

```
    MPI_Finalize();
```

```
    return 0;
```

```
}
```

⁶https://en.wikipedia.org/wiki/Parallel_computing

⁷https://en.wikipedia.org/wiki/Message_Passing_Interface

4 Using a cluster script

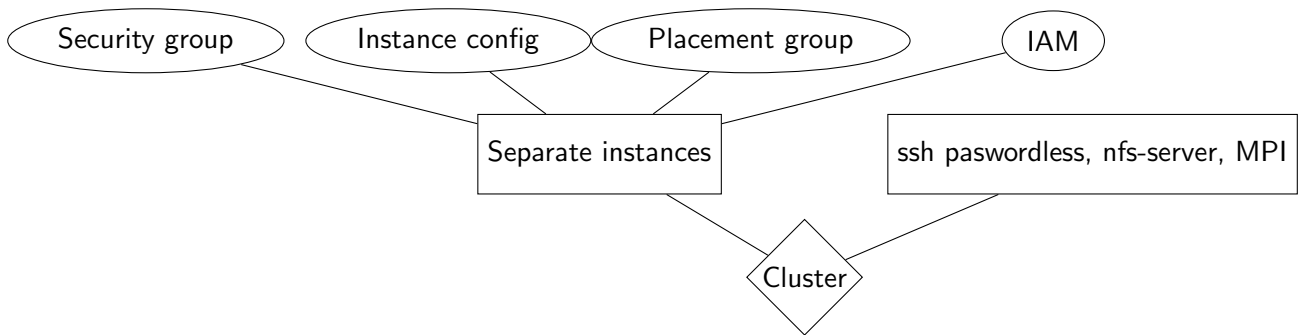


Figure 1: Schematic diagram of cluster created in this project

4.1 SSH passwordless

SSH within cluster (eg. master to node) must be setup as passwordless to achieve properly working cluster. Standard way to do this is described in **this tutorial**. However, there is an easier way if we created all AWS instances with the same key pair. The solution is *key agent forwarding* and proper ssh configuration on your local machine ⁸.

4.2 Nfs - server and sharing master's folder

NFS allows to create a folder on the master node and have it synced on all the other nodes ⁹. It reduces the problem of unnecessary redundancy. This folder can be used to store programs.

4.3 Setup for Linux distribution packages' update and install

In order to update and install all necessary packages under a clean, already installed Linux distribution (in this case Ubuntu 18.04) the following command is used:

```
$ sudo DEBIAN_FRONTEND=noninteractive apt-get -y update
$ sudo DEBIAN_FRONTEND=noninteractive apt-get -y install ...
```

"DEBIAN_FRONTEND=noninteractive" sets the default answer if an interactive prompt will appear during package update. "-y" does the same task for ordinary questions in the terminal.

5 Parallel computing - structural analysis in Elmer

To carry out parallel computation, which is the final task in this project, Elmer was chosen, which is open source software for multiphysics simulation which uses the finite element method to solve phenomena described by partial differential equations. Main advantages which are helpful in this project are the fact that Elmer is provided as Ubuntu packages repository in MPI compiled version. It is important, because the compilation process might take quite a lot of time. Installation process:

```
$ sudo apt-add-repository ppa:elmer-csc-ubuntu/elmer-csc-ppa
$ sudo apt-get update
$ sudo apt-get install elmerfem-csc
```

Great advantage of Elmer is its well documentation¹⁰.

⁸<https://www.ssh.com/ssh/agent>

⁹<https://help.ubuntu.com/community/MpichCluster>

¹⁰<https://www.csc.fi/web/elmer/documentation>