

# Tutorial 1

## Content

- EC2, S3, IAM, CLI, VPC, Subnets, Routes, Security Groups,
- Kubernetes Cluster, Minikube, Kubeadm, Kops

## AWS tasks

### Step 1: AWS CLI Configuration

An AWS classroom account has been opened for you. You must have received an invitation to register by now. Register and login to your account. The available balance of your account is \$80.00. Install `awscli` and configure your machine's terminal to be connected to your AWS account by using `awscli`. Explore the AWS documentation as required. (5 min)

[https://docs.aws.amazon.com/index.html#lang/en\\_us](https://docs.aws.amazon.com/index.html#lang/en_us)

### Step 2: Working with EC2

Create four EC2 instances using Linux AMI image and t2.micro instance type. Assign a pair (Name, A) as a tag to all instances, and finally create a key pair for SSH login in to any one of the instances using your local terminal. Launch instances and get access to instances using SSH. On your local machine, write an arbitrary simple script, transfer the application to one of the instances, run the script, log the output and then return the output to your local machine. (5min)

### Step 3: Working with S3, SSM and SNS

Create an AWS S3 bucket and upload an image file in the bucket. Set the permission for the bucket as public read, find the public link of the image and try to access it in the browser. Create another bucket, write a sample Python script to calculate the multiplication of two random valid matrixes and write the resulted matrix in a text file. Upload the Python script in the bucket. Then specify the parameters for the `AWS-RunRemoteScript` (find it at `Run a command from system management services` from the EC2 dashboard) to run the Python script on four of the instances created in step 2. For doing this, you need to configure the command to write outputs to an S3 bucket and also set the Systems Manager to send notifications about command statuses using the Amazon Simple Notification Service. Finally, terminate all instances. (10 min)

### Step 4: Working with IAM and CLI

Create a new IAM user and grant the maximum permissions. Using you IAM credentials, install and configure the AWS CLI to manage your AWS resources by command line (instead of the GUI Dashboard). Repeat the tasks from steps 2 and 3 (e.g., launch, list, and terminate instances, set key pair and security group) using the CLI. (10 min)

# Kubernetes

## Step 1: Basic Concepts, MiniKube

Explore the Kubernetes basics at

<https://kubernetes.io/docs/tutorials/kubernetes-basics/> and perform some of the interactive sessions in order to deploy a Kubernetes cluster on your local machine using minikube.

(15 min)

## Step 1: Kubernetes Cluster using KubeAdm

Create a Kubernetes cluster from scratch using the `kubeadm` cluster bootstrapping utility. Perform some cluster management tasks including creating a backup for the cluster, restoring the cluster from a backup, upgrading Kubernetes and cluster test. For doing this task, you need to create and configure four EC2 instances (one master and 3 workers) running Ubuntu 16.04, and install `kubeadm` and its dependencies (including Docker) on the master node and the workers using an SSH connection. Afterwards, initialise the master node and join all workers to the cluster. Create pods (e.g., create a deployment of the Nginx application with 5 replicas), backup and restore the state of the cluster.

**Question:** *How can you make sure that the pods are able to communicate with each other within the network?* Try to access a pod/container remotely (i.e., from your local machine). Create a web-based dashboard for your Kubernetes cluster. Refer to Kubernetes documentation whenever required. Repeat this step using `kops` instead of `kubeadm`. (15 min)

<https://kubernetes.io/docs/setup/>

### Requirements for the next tutorial:

1- Install Pyspark