MY472 - Week 3: Lab - Amazon Web Service

Pablo Barbera & Akitaka Matsuo

October 18 and 19, 2018

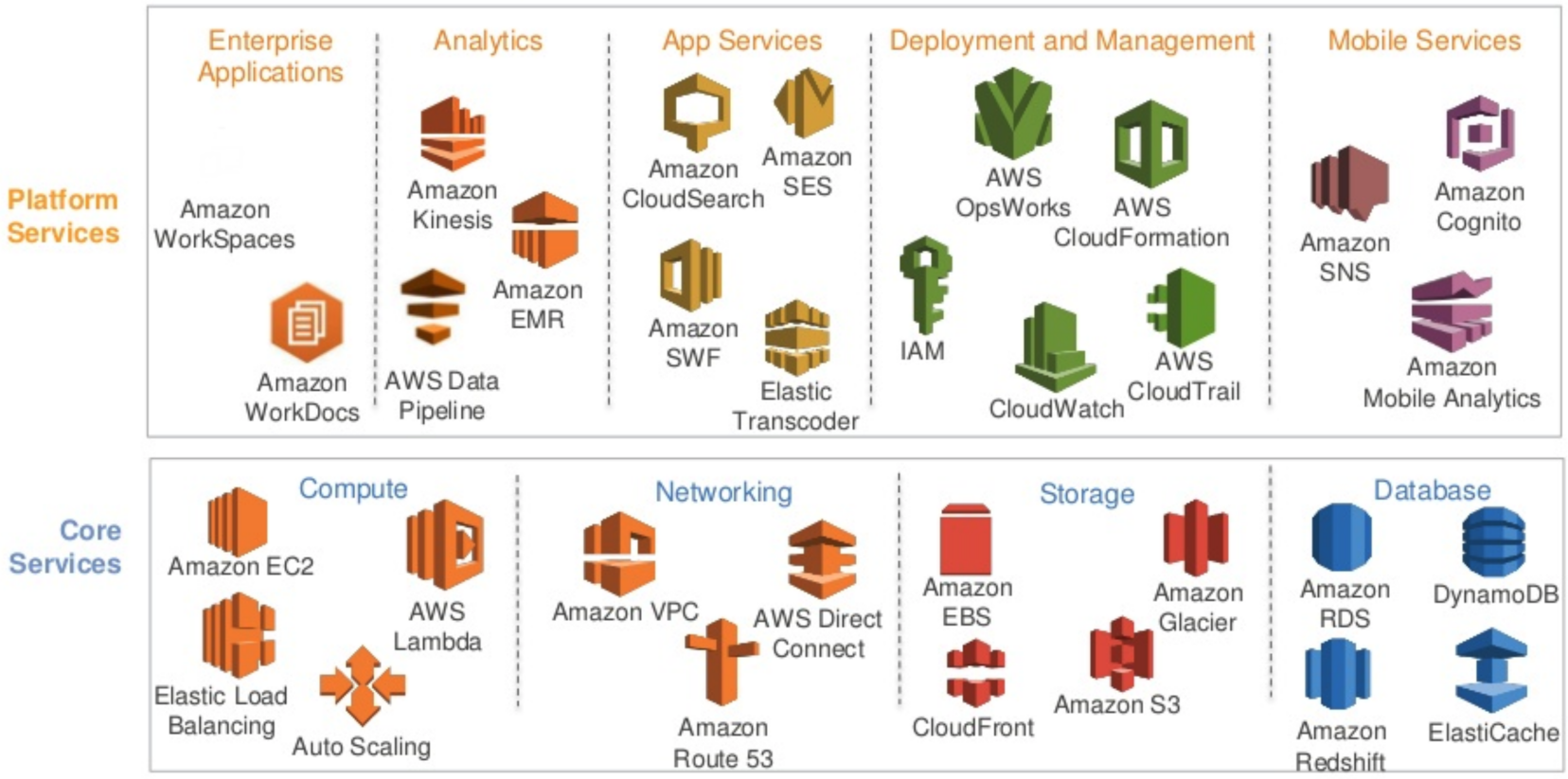
Amazon Web Service (AWS)

Some background of AWS

- Publicly launched in 2006 with EC2 and S3
- Intended to utilize the redundant computing resource at AWS
- Over three thousand services as of summer 2017

AWS Core services

Amazon Web Service



Core services (1): EC2

- EC2 = Elastic Computing Cloud
- What it is:
  - Virtual machine on a cloud
- How it works:
  - Configure the machine setup then launch
- Pricing
  - CPU usage + storage capacity

Core services (1): EC2

EC2 Configurations include:

- Machine capacity (CPU/Memory/Storage)
- Network
  - Inbound/outbound restrictions
  - "Security group"
- Machine image (AMI)
  - Operation systems (Linux/Windows)
  - Additional software
  - Customized images are available
    - Both free and proprietary
    - Examples: Rstudio

Core services (2): S3

- S3 = Simple Storage Service
- What it is:
  - Object-based storage service
- How it works:
  - Create a bucket (=folder), then put objects (=files)
- Pricing:
  - size of stored objects
    - different prices for different types of services (reliability/durability)
- Notes:
  - each object has a url and if the bucket/object is public, it's publicly accessible

AWS Educate Program

- In this lab, we use AWS Educate
  - every students on moodle must have gotten the invitation link
  - when you accept you get $50 AWS credit for MY472
  - the classroom expires in December

If you want to keep using AWS, you may want to open a real account

Free-tier

When you open an account, AWS gives you a lot of free resources for the first 12 months, including:

- EC2 (t2-micro, 750 hr/month, running one instance all the time)
- S3 (5GB)
- Lambda (1M calls/month)
- API Gateway (1M calls/month)
- RDS (750 hr/month) + DynamoDB (25GB)

https://aws.amazon.com/free

Some advertisement

Neil Prockter (LSE-HPC@IMT) and I will deliver a course "AWS Cloud Foundation", in Week 6 (Reading Week). This will help you to get a Certified Cloud Practitioner. If you are interested in getting more systematic understanding of AWS and cloud computing, please sign up from moodle. The information page is here. Registration required.

Lab Assignments

Contents

0. Go to AWS console

1. Go to AWS Educate Login
   https://www.awseducate.com/signin/SiteLogin
2. Go to MY472 Classroom
3. Open AWS console

1. Create an EC2 instance

Now we create an EC2 instance

1. Go to EC2 dashboard
2. Select region
3. Select "Launch Instance"
4. Configure the instance
   - AMI: Amazon Linux 2 AMI
   - Instance Type: t2.micro
   - Tags:
     - Key: Name
     - Value: MY472-yourname
   - Security Group:
     - Name: my472-secgroup
     - Description: my472-secgroup
     - Add rule: HTTP (everywhere)
     - Note: For simplicity, we allow SSH from everywhere, but this is not really recommended...
5. "Review and Launch"
   - Click "Launch"
   - In the popup, choose "Create a new key pair"
     - Name: my472-key
     - Download Key Pair (Filename should be "my472-key.pem". Make sure you do this, otherwise you will never be able to access the EC2 instance you just created)

2. Connect to your EC2 instance (1)

1. Go to EC2 console
2. Select “Instances” on the list
3. Check your instance state
4. Copy “Public DNS value” and paste it to somewhere
   - the value should look like:
     ec2-##-##-##-##.compute-1.amazonaws.com

2. Connect to your EC2 instance (2)

In this section, you log in to your EC2 instance with

1. Open **Terminal** (or github Terminal for windows users)
2. Change working directory to the folder you have ***.pem** file
   - `cd /path/to/the/folder`
   - in Mac, you can drag and drop the folder into the terminal window after typing `cd`.
3. Change the permission of ***.pem** file
   - `chmod 400 my472-key.pem`
4. Log in your EC2 instance via ssh
   - `ssh -i my472-key.pem ec2-user@ec2-##-##-##-##.compute-1.amazonaws.com`
   - say *yes* to everything
5. Your console before "$" now says `[ec2-user@ip-**-**-**-**]`

3. Install R

Now you install R3.4 in EC2 and check how it works (we can install newer R, but for simplicity we install 3.4).

1. Install R using amazon extras
   - In the EC2 console, enter `sudo amazon-linux-extras install R3.4`
   - This takes time but you can install fully functional R
2. Start R
   - Enter `R` in the console

4. Use R

1. Let's do very simple calculation

```
x <- rnorm(100)
y <- 2 + 2 *x + rnorm(100)
data_test <- data.frame(x, y)

model <- lm(y ~ x, data = data_test)
summary(model)
```

2. Save graphics

```
png("~/reg_plot.png", width = 600, height = 600)
plot(x, y, main = "Plot Y against X")
abline(model, col = "red")
dev.off()
```

3. Exit R

```
q()
```

5. Set up a simple website with apache on EC2

Here you install an apache server using `yum` and create a simple html file

1. Update packages: `sudo yum update -y`
2. Install apache
   - In the EC2 console, enter `sudo yum install httpd`
3. Start webserver: `sudo service httpd start`
4. Check if server is running.
   - enter public IP of your EC2 instance in your browser and see if a Test Page is shown
5. Create a simple html file to display
   - Enter `cd /var/www/html`
   - Enter `sudo nano index.html`
   - copy and paste following lines

     ```
     <html>
     <body>Hello this is my website</body>
     </html>
     ```
   - save and exit `nano`
6. Check whether the page is displayed by refreshing the browser
7. Publish the figure from the previous section
   - Copy the figure into the public folder using `cp`

     `sudo cp ~/reg_plot.png /var/www/html`
   - Check on the browser. url:

6. Using S3

In the previous section, we created a simple webpage using full-stack web-server. Instead, we will upload an html file on S3 and directly serve the file from s3. This is a cost efficient way of hosting static contents.

1. On browser, go to AWS S3 console
2. Create a bucket
   - Name: `my472-***` (since bucket names are universal, we can't give the same name to multiple buckets)
   - Region: London
   - Tags:
     - Key: Name
     - Value: my472
3. Locally Create a simple html file Using your favorite text-editor, create a simple html file and save it as `s3-test.html`: `<html> <body>This is my first S3 webpage.</body> </html>`

6. Using S3

4. Upload the file
   - Select the bucket
   - Select `Upload file`
   - Drag and drop the file `s3-test.html`
   - Choose `Grant public read access to this object`
5. Check the file
   - Click the object just uploaded
   - In `Overview` find `Link`. Click the link and check the file is displayed correctly. url should look like https://s3.eu-west-2.amazonaws.com/your-bucket-name/s3-test.html
6. (Optional) Upload other stuffs
   - You can try to upload other things (e.g. knitted `rmd`)
   - Note: Please don't forget that your free-tier limit is 5GB.

7. (Optional) Publishing more on the webpage

In this part of assignment, we knit an rmarkdown file to html.

1. Clone the forked repository with R-studio.
2. Edit the rmarkdown file `pset3-webpage.rmd`
3. knit the file to html
4. `scp` (secure copy command) to the EC2 instance `scp -i my472_key.pem pset-webpage.html \\ ec2-user@ec2-##-##-##-##.compute-1.amazonaws.com:/var/www/ht`
5. check the webpage exists
6. paste and save the url you checked in 5 into the `my_pset3_urls.rmd`
7. Do the same for S3 (upload the file on s3, give public access, and then paste the url)

Additional: Secure your account in real environment

If you decide to use AWS, you need to secure your account to avoid the disastrous amount charges on your account. In general the disaster could happen by two causes:

- Allow hackers to override your account
- Use services you didn't intend (deploy excessive resources, forget to turn off stuffs)

The followings are the minimum things you can do to prevent that

1. Secure your account

- When you open an account, the user you just created is the *root* user
- For security reasons, this account should not be used for daily work
- In the following, you secure the root user by deleting the console access credentials
- Then create a new user which you will use daily

User management

Steps to protect root user

1. Log in AWS
2. Go to IAM
3. Select `Users` on the dashboard
4. Click `Add User`
5. Delete your root access keys
    - Access keys are for accessing using AWS Command Line Interface. Root should not have CLI access.

Also, setting two factor authentication is good idea.

For more info, see documentation

User management

Steps to create a user

6. Create a user group
   - Example name: `myAdminGroup`
   - Check `AdministratorAccess`
7. Create an individual user
   - Example name: `myAdmin`
   - Check `AWS Management Console access` (not `Programmatic Access`)
   - For permission, check `myAdminGroup` (by this, `myAdmin` will have `AdministratorAccess`)
8. After creation, click download csv (we need `Password` to log-in).
9. Also save the custom login url: **Users with AWS Management Console access can sign-in at:**
   `https://************.signin.aws.amazon.com/console`

From now, we will log in the account using myAdmin user credential. This admin is a bit too powerful though, for daily business, you may want to create other users with more restricted access (e.g. EC2 + S3 only access)

For more info, see documentation

2. Set up Billing Alert

Now, we set up a billing alert so that you will get notified when the monthly charge exceeds the amount you are willing to pay.

1. Log in AWS
2. Go to Cloud Watch
3. “Create Alarm”
4. Select “Billing metrics”
5. Set the appropriate amount for “EstimatedCharges”
6. Select “NotifyMe”, put your email address
7. Log out

For more info, see documentation