**AWS BUDGET SETUP**

**My account--> my billing dashboard--> budget-->create a budget--> cost budget--> set your budget -->configure alerts-->confirm budget**

**SECTION 1) AWS FUNDAMENTALS : IAM +EC2**

**a) AWS regions and AZs**

1. **Regions :** is cluster of data center
2. Each region has many availability zone ( usually 3 , min is 2 and max is 5)
3. Each availability zone (AZ) is one or more discrete data center with redundant

Power , networking and connectivity but they are separated from each other

So that they are isolated from disaster

4) They are connected with high bandwidth

Machine generated alternative text:
AWS Region 
Sydney: ap-southeast-2 
ap-southeast-2a 
a p -southeast-2b 
ap-southeast-2c 

1. IAM: is global service **(note** : selection of region is not required)

b) IAM Introduction

1. IAM : Identity and access management
2. Root account should never be used
3. User must be created with proper permissions
4. Policies are written in JSON (JavaScript Object Notation)

Notes : Roles: we give to machine

Users : Physical person

Groups : Function( admins,devops)

Teams( engineering ,design)

Machine generated alternative text:
IAM 
Introduction 
Users 
Usually a physical 
person 
Groups 
Functions (admins, 
devops) 
Teams (engineering, 
design...) 
Contains users! 
Policies (JSON Documents) 
Roles 
Internal usage within 
AWS resources 
Defines what each of the above can and cannot do 

1. IAM has a global view means when we create users, group or policies it will applicable

Across all regions

6) Permission are governed by Policies (JSON)

7) MFA( Multifactor authentication can be set up)

8) IAM has predefined managed policies

Machine generated alternative text:
IAM 101 Brain Dump 
• One IAM User per PHYSICAL PERSON 
• One IAM Role per Application 
• IAM credentials should NEVER 
RED 
• Never, ever, ever, ever, write I -Men ials in code. EVER. 
• And even less, NEVER EVER MIT YOUR IAM credentials 
• Never use the ROOT account except for initial setup. 
• Never use ROOT IAM Credentials 

NOTE:

Create IAM users and give them only the permissions they need. Do not use your AWS root account for day-to-day interaction with AWS, because the root account provides unrestricted access to your AWS resources.

1. We can add user to group and it will inherit the permission of group

b) EC2 Introduction

An **EC2** instance is nothing but a virtual server in **Amazon Web services** terminology. It stands for **Elastic Compute Cloud**. ... An on-demand **EC2** instance is an offering from **AWS** where the subscriber/user can rent the virtual server per hour and use it to deploy his/her own applications

**Amazon Elastic Compute Cloud** (**EC2**) is a part of [Amazon.com](https://en.wikipedia.org/wiki/Amazon.com)'s [cloud-computing](https://en.wikipedia.org/wiki/Cloud-computing) platform, [Amazon Web Services](https://en.wikipedia.org/wiki/Amazon_Web_Services) (AWS), that allows users to rent [virtual computers](https://en.wikipedia.org/wiki/Virtual_computer) on which to run their own computer applications. EC2 encourages scalable deployment of applications by providing a [web service](https://en.wikipedia.org/wiki/Web_service) through which a user can boot an [Amazon Machine Image](https://en.wikipedia.org/wiki/Amazon_Machine_Image) (AMI) to configure a [virtual machine](https://en.wikipedia.org/wiki/Virtual_Machine), which Amazon calls an "instance", containing any software desired. A user can create, launch, and terminate [server](https://en.wikipedia.org/wiki/Server_(computing))-instances as needed, paying by the second for active servers – hence the term "elastic". EC2 provides users with control over the geographical location of instances that allows for [latency](https://en.wikipedia.org/wiki/Network_latency) optimization and high levels of [redundancy](https://en.wikipedia.org/wiki/Redundancy_(engineering)).

NOTE: the application we deploy in EC2 called instance

Note : we can launch multiple instance on AMI

1. it mainly consist in the Capabilities of :

Renting Virtual Machine(**EC2**)

storing data on virtual drives (**EBS - Elastic Block Store**)

distributing load across machine(**ELB -Elastic Load Balancing**)

scaling the service using an auto-scaling group (**ASG**)

2) Creating EC2  **(IMP work on amazon linux 2 AMI)**

a)Launch EC2 instance

b) Choose AMI**(amazon machine image**) this is basically operating system launch on server

c) Choose Instance type (t2.micro)

**Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications**

d) Configure instance details

**Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.**

No. of instances (For ASG)

Choose VPC

Choose Subnet

Assign Public IP

Select Placement Group

Select Capacity Reservation

Choose IAM Role

Set Shutdown behavior

Enable termination protection

Enable Cloud watch monitoring

Decide Tenancy

e) Add Storage

**You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes.**

f)Set Tags

**A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.**

**A copy of a tag can be applied to volumes, instances or both.**

g)Configure Security Groups

Decide ports and IPs to allow access

**A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below**

Review & Launch

h)Select Key-Pairs

**Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications**

download the private key file Save the (\*.pem) file securely

NOTE: we can right click on instance and terminate it

1. **SSH AND EC2**

**Instance or virtual server**

**SSh using windows ( to login into ec2)**

**Secure Shell** (**SSH**) is a software standard to support encrypted data transfer between two computers. It can be **used to** support secure logins, file transfers or general purpose connects.

It allows to control a remote machine all using command line

**SSH using windows 10**

Machine generated alternative text:
EC2 Machine 
Linux 
Public IP 

**STEP 1)Convert .pem to .ppk using PuttyGen**

( click file -->load private key(downloaded .pem file while creating ec2 instance)--->select all files->save Private key ---Ec2Tutorial.ppk (putty private key file)

**STEP2) Login to EC2 instance using public IP and .ppk file using putty**

note : public ip details get from ec2

**use putty for login**

a) host name : ec2-user@35.180.254.179 ( user name ===ec2-user)

b) link .ppk file --->connection-ssh-auth--private key file for authentication-browse-select ppk file-session-save-open (hurrah we are now inside machine)

Machine generated alternative text:
Using username "ec2—user" 
Authenticating with public key "imported—openssh—key" 
Amazon Linux 2 AMI 
https : / / aws . amazon . com/amazon—Iinux—2/ 
Lec2—user@ip—172—31—38—219 —J $ who am i 
ec2—user pts/ 0 
2020-10-01 06:40 (128.107.241.179) 
Lec2-user@ip-172-31-38-219 

1. SSh using windows 10

Note : The user log onto ec2 machine is always ec2-user

Ssh -i path of .pem file ec2-user@public ip

**ssh -i C:\Users\saurakes\Desktop\niagara\_doc\aws\_stephane\Ec2Tutorial.pem ec2-user@35.180.254.179**

Machine generated alternative text:
PS C:XUsersXsaurakes) ssh 
C:XUsersXsaurakesXDesktopXniagara docxaws stephaneXEc2Tutoria1.pem ec2-user@35.18a.254.179 
The authenticity of host '35.188.254.179 (35.180.254.179)' can't be established. 
ECDSA key fingerprint is SHA256:sirowjfpKKawunfzZbxvycni88rGwjGDeg6ØGYEaT0. 
Are you sure you want to continue connecting (yes/no)? yes 
Warning: permanently added '35.183.254.179' (ECDSA) to the list of known hosts. 
Last login: Thu Oct 
1 2ø2a from 128.137.241.179 
Amazon Linux 2 AMI 
https : / / aws . amazon . com/amazon-linux-2/ 
Lec2-user@ip-172-31-38-219 M$ who am i 
ec2-user pts/a 
2020-10-01 Ø7:36 (128.1Ø7.241.179) 
Lec2-user@ip-172-31-38-219 

1. EC2 Connect

Action-connect ---- behind the scene aws will upload key to ec2 instance that allow me to connect temporarily

NOTE: ec2 connect will not work if we removed inbound rules from security group

So to work ec2 connect we need ssh port (security-add inbound rule)

Ssh port 22:

Resources --anywhere ( means for any ip)

Click ec2-connect now it will work

1. Security Group Deep Dive

Security group control inbound and outbound traffic of ec2 instance

Machine generated alternative text:
www 
Inbound traffic 
Outbound traffic 
EC2 Machine 

suppose we are not able to login in ec2 machine then check security bound inbound rules

security group acts as firewall on ec2 instance

**They regulate:**

Access to port

authorized ip ranges IPv4 and IPv6

control of inbound network ( from other to instance)

control of outbound network( from other to instance)

Machine generated alternative text:
Security Groups 
Diagram 
Sec u Group 
Inbound 
Filter IP / Port with Rules 
EC2 Instance 
IP XX.XX.XX.XX 
Security Group 1 
Outbound 
Filter IP / Port with Rules 
Port 22 
Port 22 
An Port 
Your Computer - IP XX.XX.XX.XX 
(authorised port 22) 
Other computer 
(not authorised port 22) 
www 
Any IP — Any Port 

Security group can be attached to multiple instance

All inbound traffic is blocked by default

All outbound traffic is authorized by default

Machine generated alternative text:
Referencing other security groups 
Diagram 
Security 
EC2 Instance 
Group 2 
IP XX.XX.XX.XX 
(attached) 
Security 
EC2 Instance 
Group 1 
IP XX.XX.XX.XX 
(attached 
Security 
EC2 Instance 
Group 3 
IP XX.XX.XX.XX 
(attached) 
Port 123 
Port 123 
Port 123 
EC2 Instance 
IP xx.xx.AA.AA 
Security Group 1 
Inbound 
Authorising Security Gro 
Authorising Security Group 
people have written a note here. 

g) private vs public vs elastic IP

Aws support both IPv4 and IPV6

IPV4-[0-255].[0-255].[0-255].[0-255]. ( 3.7 billion address)

**Public ip :**

machine can be identified on the internet

**Private ip:**

machine can only be identified on private network only

Machine connect to www using a NAT + internet gateway(a proxy)

**Elastic ip**

When we stop and start ec2 instance it can change it public ip

If you need to fixed public ip for an instance you need an elastic ip I

An elastic ip is a public ipv4 you own as long as you don’t delete it

you can attach it to one instance at a time

Machine generated alternative text:
Private vs Public IP (IPv4) 
In AVVS EC2 - Hands On 
• By default, your EC2 machine comes with: 
• A private IP for the internal AWS Network 
• A public IP,for theVvWW. 
• When we are doing SSH into our EC2 machines: 
• We can't use a private IR because we are not in the same network 
• We can only use the public IR 
• If your machine is stopped and then started, 
the public IP can change 

very Imp : if we start and stop ec2 instance public ip will change but private ip will remain same

**Question how we make sure public ip remain same for ec2 stop and start for this will use elastic ip**

**Associate elastic ip** :-Network-security-elastic-ip -associate elastic ip with ec2 instance --- choose ec2 instance

Once elastic ip is associated with ec2 instance then public ip and elastic ip will remain same

Now even we stop ec2 instance public ip will remain same

**Release elastic IP :**

right click on ec2 /networking/dissociate elastic ip address

As soon as we release elastic ip will get new public ip address

Important : go to network/elastic ip /action/release ip address

1. Launching an apache server on EC2 and installing it

Will install apache web server on ec2 to display a page

We will create index.html that will show hostname of our machine

**Powershell**

ssh -i C:\Users\saurakes\Desktop\niagara\_doc\aws\_stephane\Ec2Tutorial.pem ec2-user@15.188.50.239

**Step 1)**[ec2-user@ip-172-31-38-219 ~]$ **sudo su** ( it will help to gett root and execute any command)

**Step 2)**[root@ip-172-31-38-219 ec2-user]# **yum update -y** (it will update all , -y allow to update all without prompting yes)

**Step 3) yum install httpd -y**

**Step 4) start service : systemctl start httpd.service**

**Step 5) enable service : systemctl enable httpd.service**

**Curl localhost:80** systemctl enable httpd.service

* + [**http://15.188.50.239:80**](http://15.188.50.239:80)

NOTE: any time timeout issue it probably with security group

NOTE: after adding security group above url will work

Machine generated alternative text:
A 
Not secure 
15.188.50.239 
Test Page 
This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at 
pal 
this site is working properly. 
If you are a member of the general public: 
The fact that you are seeing this page indicates that the website you just visited is 
either experiencing problems, or is undergoing routine maintenance. 
If you would like to let the administrators of this website know that you've seen this 
page instead of the page you expected, you should send them e-mail. In general, 
mail sent to the name "webmaster" and directed to the website's domain should 
reach the appropriate person. 
For example, if you experienced problems while visiting www example.com, you 
should send e-mail to "webmaster@example.com". 
If you are the website administrator: 
You may now add content to the directory /var/www/html/. Note that until you do so, 
people visiting your website will see this page, and not your content. To prevent this 
page from ever being used, follow the instructions in the file 
/ etc/httpd/conf.d/welcome.conf. 
You are free to use the image below on web sites powered by the Apache HTTP 
Server: 
wered by 
2.4 
APACHE 

Note : we can add content to /var/www/html/.

**[root@ip-172-31-38-219 ec2-user]# echo "hello world" > /var/www/html/index.html**

Now we see first web page on machine which is awesome

Machine generated alternative text:
C 
hello world 
A Not secure 
15.188.50.239 

H ) EC2 User Data (Automate it)

It is possible to bootstrap our instance using an EC2 User data script

Bootstrap means launching command when machine starts

That script is only run once at the first start

EC2 user data is used to automate boot tasks such as:

* Install updates
* Install software
* downloading common files from internet
* anything we think of

EC2 user data script runs with root user

**EC2 user data hands on**

**Requirement : ec2 instance has an apache http server installed on it to display a simple web page**

**IMPORTANT**

* Add **User Data Script** while creating EC2 instance under **Advanced Details** on **'Configure Instance Details**' page

we need to paste whole script there

#!/bin/bash **( first line add it very important)**

# install httpd (linux 2 version)

sudo su

yum update -y

yum install httpd -y

service httpd start

service httpd status

echo "hello world from $(hostname)" > /var/www/html/index.html

NOTE: even terminate instance security group will still there will still can use it

<http://35.180.35.138/>

hello world

**NOTE:**

Anything that ends with a letter is an AZ

1. EC2 Instance Launch Type

**1)EC2 On Demand**

* Pay for what you use(billing per second, after the first minute)
* Has the highest cost because it is on demand
* No long term commitment

**2)Reserved Instances**

* Long Workload (>= 1 year)
* 75% discount compared to On-Demand
* Pay upfront for long term commitment
* Reserve for 1 - 3 years
* Reserve a specific instance type (Eg.: x4-large)

**3)Convertible Reserved Instances**

* Long Workload with flexible instances
* Can change instance type
* 54% discount compared to On-Demand

**4)Scheduled Reserved Instances**

* Launch within reserved time window
* Use when you need (Day, Week, Month) (Eg.: Every Sat-Sun)

5)**Spot Instances**

* Short Workload
* Cheap
* Can lose instances
* Up to 90% discount compared to On-Demand
* Get the instance by bidding
* Use the instance till bid amount is above the spot price
* Price depends on demand and offers
* Instance lost withing 2 mins notification after spot price crosses bid amount
* Typically used for Batch Jobs, Big Data Analysis which are resilient to failures

6**)Dedicated Hosts**

* Book entire physical server
* Control instance placement
* Visibility to underlying socket, processor cores, hardware, etc.
* Allocated for a 3 year period
* Much expensive
* Useful in cases of Complicated Licensing model

**7)Dedicated Instances**

* No other customer will share hardware
* May share hardware with other instance under same account

Machine generated alternative text:
Which host is right for me? 
• On demand: coming and staying in resort 
whenever we like, we pay the full price 
• Reserved: like planning ahead and if we plan to 
stay for a long time, we may get a good 
discount. 
• Spot instances: the hotel allows people to bid 
for the empty rooms and the highest bidder 
keeps the rooms. You can get kicked out at any 
time 
• Dedicated Hosts: We book an entire building 
of the resort 

1. EC2 Elastic Network Interface(ENI)

Machine generated alternative text:
Elastic Network Interfaces (ENI) 
• Logical component in a VPC that represents a 
virtual network card 
• The ENI can have the following attributes: 
• Primary private IPv4, one or more s 
• One Elastic IP (IPv4) per private I 
. One 
Public IPv4 
• One or more security groups 
• A MAC address 
• You can create ENI independently and attach 
them on the fly (move them) on EC2 instances 
for failover 
• Bound to a specific availability zone (AZ) 
Availability Zone 
Etho — primary ENI 
192.168.0.31 
EC2 
Ethl — secondary ENI 
192.168.0.42 

K ) Good things to know about EC2

* EC2 Pricing (Only if the instance in Running)
  + Per Hour Pricing depending on
    - Region (Mumbai)
    - Instance Type (t2.micro)
    - Launch Type (On-Demand)
    - OS (Linux)
  + Per second billing (after 60 seconds)
  + Other chargable factors
    - Storage
    - Data Transfer
    - Fixed IP
    - Load Balancing
* AMI (Amazon Machine Images)
  + Predefined
    - Ubuntu
    - Fedora
    - RedHat
    - Windows
    - Etc.
  + Can be customised using EC2 User Data Scripts
  + Custom AMIs can be used
    - Pre-installed packages
    - Faster Boot time (No User Data Scripts)
    - Pre installed monitoring and network tools for enterprise
    - Control maintenance and updates over time
    - Configure LDAP out-of-the-box
    - Install application before machine boot on all machines (During auto-scaling)
    - Someone else's exported AMI
  + Custom AMIs are region specific
* EC2 Instances Characteristics
  + RAM (Type, Amount, Generation)
  + CPU (Cores, Type, Make, Frequency, Generation)
  + I/O (Disk Performance, EBS optimisations)
  + Network (Bandwidth, Latency)
  + GPU (Present or Not?)
  + Permutations results intomore than 50 instance types
  + Summary : [https://ec2instances.info](https://ec2instances.info/)
  + R/C/P/G/H/X/I/F/Z/CR - Best in their characteristic
  + M - Balanced (Good overall, Great in nothing)
  + T2/T3
    - OK CPU overall
    - When Spike in processing, CPU bursts (VERY GOOD processing)
    - In burst mode, uses "burst credits"
    - If all credits are used, CPU becomes BAD
    - If machine stops bursting, credit accumulate over time
  + T2 Unlimited
    - Unlimited credits for high cost
* EC2 Checklist
  + SSH to EC2
  + .pem to .ppk (PuTTY)
  + change permission of .pem (0400 on Linux)
  + use security group properly
  + public vs private vs elastic IP
  + User Data Scripts to customize instance
  + Custom AMI to enhance OS
  + EC2 billing is per second (after 60 seconds)