

12/3

Cloud computing

① Basic / traditional hosting of service.

- * Buy a stack of servers
- * keep peak traffic
- * Monitoring & Maintenance.

② Disadvantages:

- * Expensive
- * Troubleshooting
- * Traffic: servers will be idle.

③ Cloud Advantages:

- * No more buying servers.
- * Scalability.
- * Cloud provider manages all.

Cloud computing definition:

It is to use remote servers available on internet to store, manage & process rather than a local server or your personal computer.

Cloud Models :



Service Model

(IaaS, PaaS, SaaS)

Deployment Model.

(Public, Private, Hybrid)

IaaS : [Virtual data center]. (AWS, Azure.)

Through virtualization provider gives cloud computing infrastructure, including servers, networks and OS & Storage.

Pros

- * Resources available as service
- * Cost varies depending on consumption
- * Services are scalable.
- * Multiple users on single piece of hardware.
- * Reliable & flexible.

Cons

- * Security
- * Legacy apps → May not be supported
- * Training.
- *

platform as service: (only user interface)

- * You will not bother about the OS.
- *

Global infra

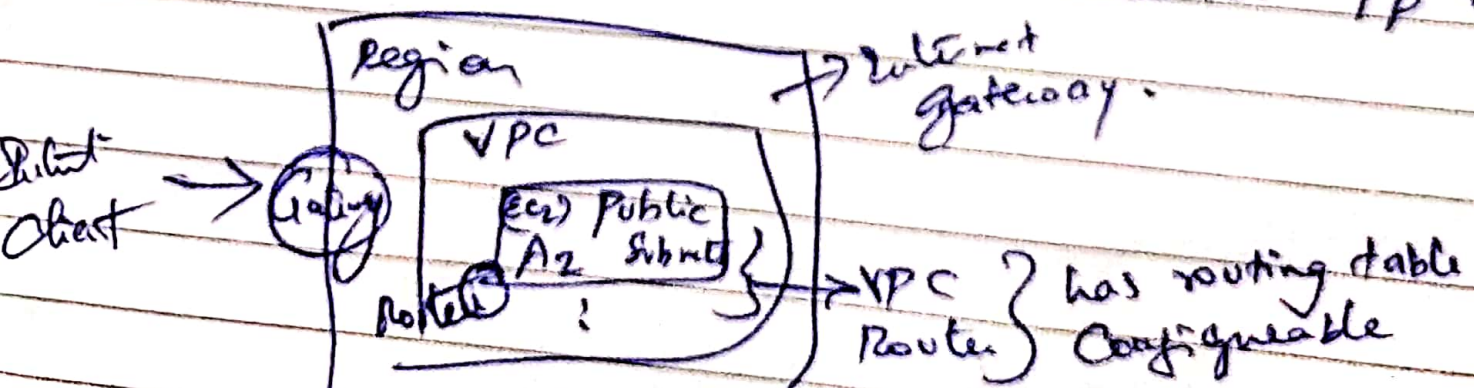
Region (23)

- ↳ Availability zone
- ↳ Edge location
 - ↳ Regional Edge cache
 - ↳ Global network

Virtual

Private cloud:

logically isolated portion of AWS cloud
You can create own network with IP range

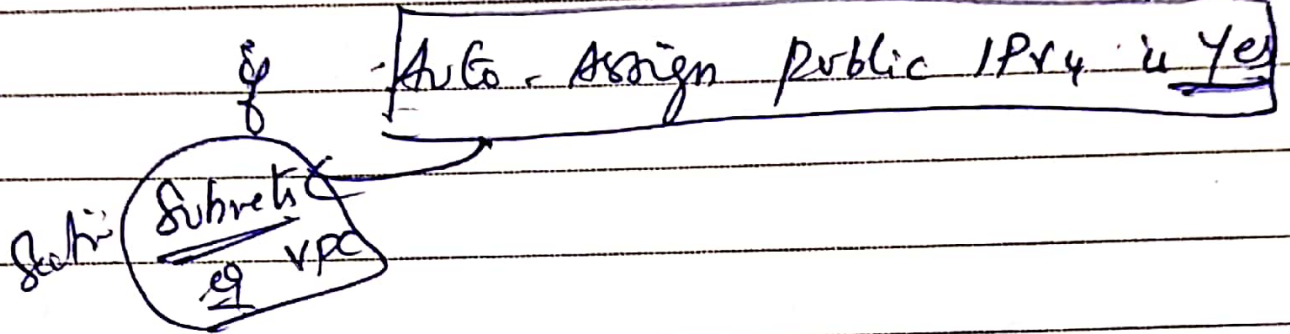


2nd in EC2 → User installs an operating system



IPV4

2nd proc of the ~~IP~~ address from the VPC



Route 53: (DNS service)

It ~~is~~ incurs cost.

→ Domain Registration

(depends on domain - cost)

→ (Hosted) ←

EC2: Elastic Compute cloud

(foundational service)

→ Launch & Scale service instances

→ 750 hours / month.

EBS (Elastic Block Store)

underlying storage of EC2 instance

EB51

volumes : Can choose SSD, HD or Magnetic

[Linux servers]

→ sudo su → super user

yum update -y (to update latest)

PuTTY

→ copy the IPv4 address

putty gen : Choose the pem file & convert to ppk file.

PuTTY : Host: ec2-user@(IPv4 address)

↓ Connection

→ SSH

→ Auth (choose ppk file)

(Open)

Aug -> Core Config

70 Add Windows instances

- ① Copy the .pem to txt file.
- ② Connect to the RDP (as windows)
 - ↳ Decrypt the txt file.
 - to get password
- ③ Give IP to RDP & you can add to login.

Security Groups:

firewalls that applied on instance level. Traffic in/out on instance.

ICMP Ping between instances →
Inbound traffic is always allowed.
If both instances are in same
Security group.

Metadata :

- ① curl `http://169.254.169.254/latest/meta-data`
- ② file Based:
`wget https://s3.amazonaws.com/ec2metadata/ec2-metadata`

`ls` - list the files.

↳ You should see a bare file

`ls -la ec2-metadata`

To see what level of permission I have

`chmod u+x ec2-metadata`

↳ give executing permission.

`ec2-metadata -help` [list down flags]

Run the tool

(eg) `ec2-metadata -P` (public-hostname)
`-z` (availability zone)

Status checks & monitoring



System checks

(Problem with system
which AWS needs
to check)



Instance checks

(On the instance
which you need to
check)

Cloudwatch!



Basic

Detail
(pay)

1-min metric collection.

eg)

create an alarm for terminating an
instance when the CPU utilization is
more than 80%.

```
sudo amazon-linux-extras install epel-y
```

```
sudo yum install stress -y
```

```
stress --help
```

```
stress -c 8
```

(Assign 8 workers which
makes more CPU utilization)

Refresh the log screen to see the instance
is terminated.

IP addresses.

Public, Private & Elastic ~~Subnets~~

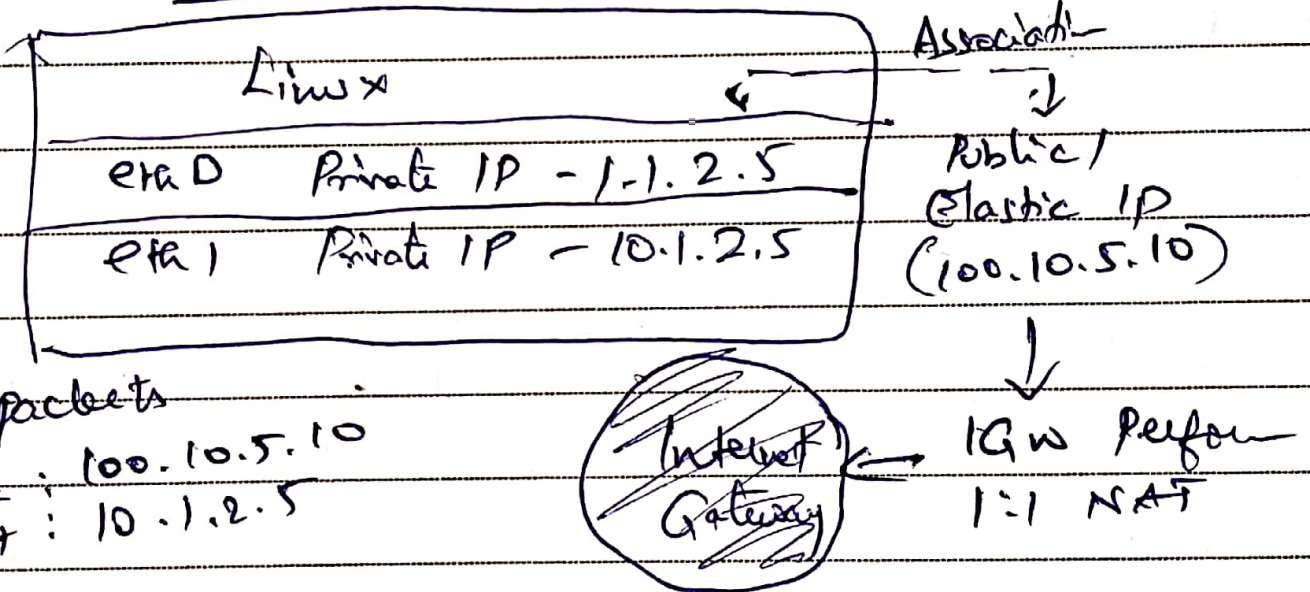
Public - lost when instance stopped, NO cost

Private - retained " " " , Used in

(ifconfig?) Public & private Subnets.

Elastic - Static public IP address, Charge involved

ECC instance



Linux OS doesn't know anything abt the Public / Elastic IP. AWS perform routing through. (IGW).

Boston Subnets

(Creating Private Subnets)

login to vpc console:

→ create route table.

[Public Subnets: Auto Assign IPv4 public to Default Route will be yes]

ssh-add -c "keypair.txt"

eval 'ssh-agent -s'

ssh-add -c "keypair.txt"

ssh (-A) ec2-user@ [public-ip]

→ Port forwarding

- ① login vpc → create Route table
- ② create Private Subnets.
(IPv4 address get from IP Subnet Calculator)
- ③ Associate Route table with Private Subnets.
- ④ Create 2 instances one with public subnet & another with Private Subnet.
- ⑤ Access Public Subnet

NAT Instance & NAT Gateway:

NAT Gateway needed an elastic IP. to configure one to a public subnet so that a private subnet can configure access to public gateways:

EC2 placement groups:

- 1) Cluster: Packs instances close together inside AZ for low latency.
- 2) Partition: Spreads your instances across logical partitions such as groups of instances in one partition do not share the underlying h/w.
- 3) Spread: Strictly places a small group of instances across distinct underlying hardware to reduce correlated failures.

Pricing Models:

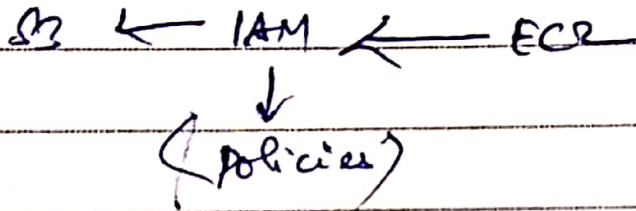
<u>On-Demand</u>	<u>Reserved</u>	<u>Spot</u>
① NO fee upfront	options: no upfront partial " all "	NO fee upfront
②	charged by hr/second	
③ NO commitment	1 yr or 3 yr	NO commitment
④ Ideal for Short term or unpredictable unpredictable or cheap costload	Ideal for steady state & predictable usage	Ideal for cost- sensitive, compute intensive use case that can withstand interruption

<u>ENI</u>	<u>ENA</u>	<u>EFA</u>
(Elastic Network Interface)	(Elastic Network Adapter)	(Elastic Fabric Adapter)
Basic adapter type	Good for cases	High Performance
When no high-performance requirements.	require higher bandwidth & low latency	Computing - MPI & ML use case
can use with all instance types.	Supported for HVM only	can be used with with all instance types

Elastic Load Balancing & Auto Scaling

Amazon S3

IAM roles



S3: <http://<bucket>.S3.<Region>.amazonaws.com>

<http://S3.<Region>.amazonaws.com/<Bucket>>

Log in to EC2 instances

① `aws s3 ls` - This should list down the folders.

② `aws configure` - Ask for secret & Access IDs to connect to AWS.
(But get stored in system).

③ `cat .aws/credentials` - Can see AWS credentials.

④ `rm -rf .aws/credentials` - Remove file.