Delete = 1) Template
2) AMI's $\xrightarrow{\text{Action}}$ Deregister
paid $\longrightarrow$ 3) snapshots $\longrightarrow$ Actions $\longrightarrow$ delete

15-6-24.

Auto scaling -

1) SNS -

Services $\longrightarrow$ App^n integration $\longrightarrow$ Create topic

create Subscriber -

2) create EC2 instance
for
custom AMI   (Allow HTTP, ssh)

3) connect it with instance using putty.
   - sudo yum install httpd wget -y
   - cd /var/www/html/
   - sudo wget (website link).
   - sudo systemctl start httpd.
   - _____ " ___ enable httpd

create highly avialable architecture.

VPC lattice - improve performance.

ii) AMI -

select instance → Actions → creat image & template
create image.

5) Delete EC2 instance.

6) Lanch template → create launch template

tags ← Instance ← select AMI ← Give name
type, SG

↳ Launch template

★ ASG (Auto scaling Groups)
↳ EC2 auto scaling → Region dependent
↳ Horizontal scaling.

Auto scaling groups → create ASG → Give grp name

select launch
choose instance ← Next. ← template
launch options

↳ N/W → Next → config. Adv option
(default vpc, select all
subnets)

No load balancer

Health check ← No vpc lattice
(30 sec)

→ Addition settings → Next → config. grp size & scaling
Desired capacity (1)
(Enable grp matrices
collect within cloud watch
Instance maintenance
Scaling limits
Next ← Policy ← No scaling (min-1, max-3)
(No policy) policy
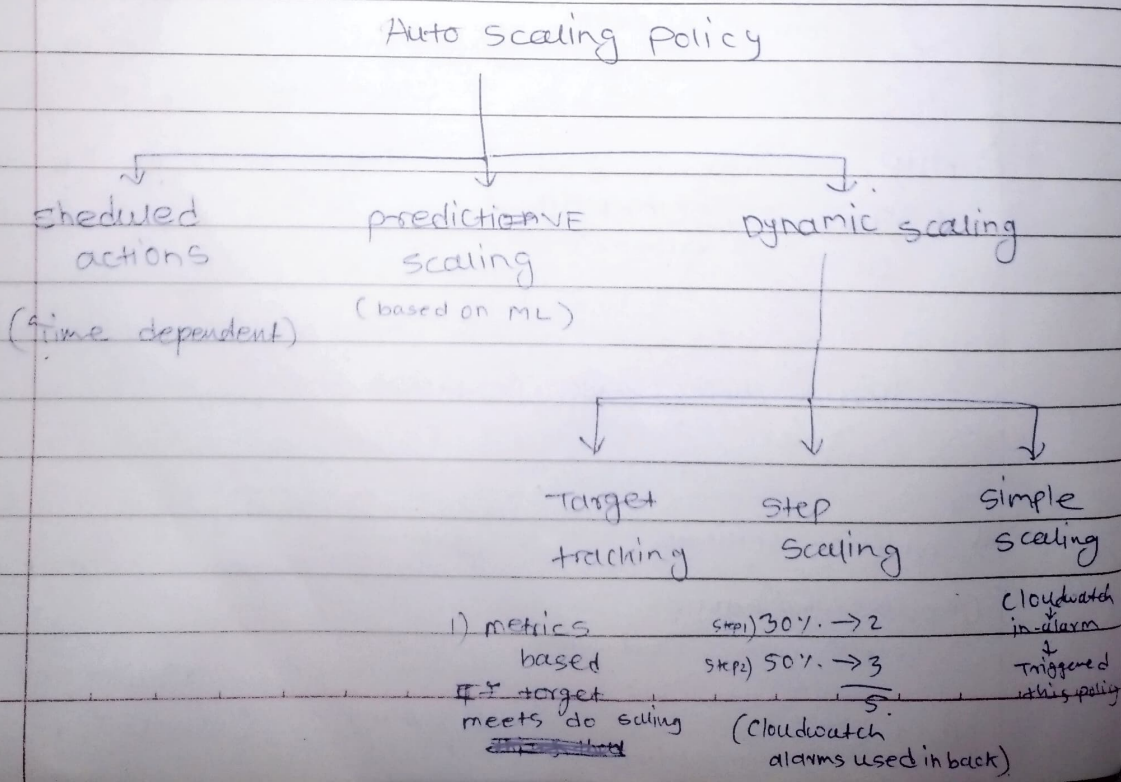
Scale in & scale out → increase instance

↓ decrease
instance
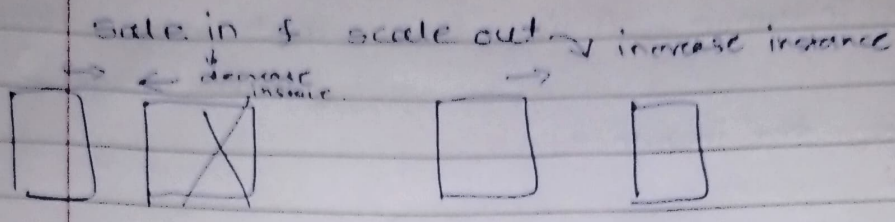
Add Notification → Select SNS Topic → Event type:
(select on)

copy public ipv4 → paste in browser.

Auto-scaling groups
check - 1) details
2) Activity
3) Instance refresh.

Auto Scaling Policy

Sheduled actions   predictionAVE   Dynamic scaling
                          scaling
(Time dependent)   (based on ML)

                    Target   Step   Simple
                    tracking   Scaling   scaling

                    1) metrics   step1) 30% → 2   cloudwatch
                       based   step2) 50% → 3   in-alarm
                    If target   ↓
                    meets do scaling   s.   Triggered
                                    (Cloudwatch   this policy
                                    alarms used in back)

EC2 → Auto scaling grp → Create
create dynamic scaling policy → TYPE
simple scaling

Cloudwatch ← Give name
alarm (scale-out)

Cloudwatch — in-alarm → select metric → EC2

By auto scaling group
Threshod ← Specify metrics.. ← (Paste name in sorch)
Static statistic (max) (select CPU Utilizati)

↳ Greater → 25 → create Next → select SNS
topic
↓
Greeter alarm

→ Select scal-out → Take Action → create
(25 > 25) Add 1
(wait ≠ 30 sec)

Also createn for scale in Sudo skill (PID value)
policy

To stop
Kill the process

Select instance → putty → yes > /dev/null &
top

monitor
cloud watch → scale-out alarm

Delete — 1) Auto scaling grp.
2) Alarms

3) SNS
4) AMI
5) Snapshot.
6) Template (optional)

free.