

EC2 (Virtual Server in the Cloud)

Instances Running

Launch Instances

Name: MyWindow

Image for OS → Search (Windows)

choose free tier eligible → Windows 2020

Key Pair (Login) → Create New

Name: Windows19/07

Type: RSA

Private key file format: .pem
Privacy Enhanced Mail

[Create]

Network Settings

Auto - assign public IP (Enable)

Firewall - Create Security group

- Allow RDP traffic from → Anywhere
- Allow HTTPS from the internet
- Allow HTTP from the internet

Edit → Add → Type: All TCP,
Range: 0 - 65535, Type: Anywhere

Configure Storage

30 GiB

Summary

No. of instances (Machines): 1

[Launch]

Instance State

Stop Instance

Connect to instances → RDP →
Download Remote Desktop file

Password → Upload Private
key file → .pem file

Windows
Server

Elastic Block Store - Volumes

Create Volume - Type: GP3-SSP(gp3)

Size: 5, IOPS: 3000, Throughput: 125

Windows
Volume

Create Volume

Click New Volume

Actions → Attach Volume

1 instance (Windows or Linux)

Device Name: xvdb

Attach

Check EC2 Dashboard → RDP

Enable Volume This PC → ✓ Click →

Manage → Server Manager → Tools →

Computer → Disk Management → Disk (New Volume) → ✓ Click → Online → Initialize Disk → Click Unallocated → New Simple Volume → Next...

Linux

df -h

lsblk

Create Volume → Device Name: xsd1sde

df -h

lsblk → (New xvde)

Sudo su root or Sudo su

mkfs -t ext4 /dev/xvde

mkdir newvolume

mount /dev/xvde newvolume

On-Premise - Very difficult, Physical Infrastructure

Auto Scaling - Virtual infrastructure

* CIDR - Classless Inter-Domain Router

Classful

Server
Linux → Public IP (copy) →
WinSCP (Use Paste)

@ apt-get install apache2
or
apt-get install httpd (NA)

① apt-get update.

cd /var/www/html
remove
directly access the Linux root file system

ls

vi index.html

cd (root)

chmod 777 /var/www/html

ls -l /var/www/html/ directory

google free website template (Download)

Win SCP

① Make sure Linux is started

② New Tab → Public IP → Username: ubuntu
Password: → Advanced → SSH → Authentication
→ Private Key File → Browse → Upload PPK file
→ Login

③ home & tick → var/www/html

④ Copy downloaded web folder

Windows Web Host

① This PC → Manage → Add Role
Features → Next → Server Roles →
Select MS → Web Server (IIS) →
Then Install

② This PC → C → inetpub → www
root (Web file new)

VPC ~~Cloudless~~ Inter Domain Routing (CIDR)
Start 32-1, 31-2, 29-4, 28-8 (172.16.0.0 → 172.16.0.255)
~~32-1, to 32-2,~~
32-1, 31-2, 30-34 to 19-

VPC Virtual Private Cloud
CIDR - Cloudless Inter Domain Routing
Start 32-1, 31-2, 30-4 to 19-^{8, 19, 2} XDAK

CIDR to IP

172.16.0.0/24

① VPC → Create VPC → VPC only →

Name: Nivi-My-VPC

IPv4 CIDR block: Manual input

IPv4 CIDR: 10.0.0.0/16

Tenancy: Default Create

② Subnet inside VPC

Subnets → Create Subnet → Choose (Nivi)

Subnet name: Nivi-My-Subnet-PUB

IPv4 Subnet CIDR block: 10.0.1.0/24
Create Subnet

Name: PR1

: 10.0.2.0/24

③ Internet Gateways

Create IG → Nivi-My-IGW Create

Actions → Attach to VPC → Nivi-My-VPC

Attach

④ Route Tables

Create → Nivi-My-RT-PUB

VPC → Nivi-My-VPC Create

⑤ Subnet associations

Edit → Pub → Save

⑥ Routes → Edit → Add route → Target: IG → Nivi-IGW
Dest: 0.0.0.0/0 Save



Route → Create → Name: Nivi-RT-PUB

~~Edit route~~

Security → Security Groups → Create →

Name: Nivi-My-SG-PUB

Des: "

VPC: Nivi-VPC

Inbound rules → Add rule → Type: All TCP

Source Type → Anywhere

SSH, RDP, HTTP → Anywhere Create

Another SG for Private

Name: Nivi-My-SG-PRT

VPC: Nivi-VPC

Inbound rules → 10.0.1.0/24 - All TCP

Create

ECD → Ensure Location → Create New

Instances → Name: Nivi-PUB

Public Key Pair → New (Nivi-My-KP-PUB) Edit

Network Settings → Nivi-My-VPC

Subnet: Public, Auto-assign public IP

Enable, → Select existing Security group
Public Launch

Private Subnet: Private, Disable,

Select existing Security group: Private Launch

Connect → Public Instances → RDP →
Password

NAT - Network Address Translation

Win+R metsc

Username: Administrator

Password:



VPC → NAT gateways → Create NAT → Nir
Name: Nir-NAT, Subnet: Nir-PUB,
Connectivity: Public, Allocate Elastic IP, enable

Create

NAT gateway →

NAT

VPC Route Table (Prov)

Route 2

Edit

NAT GW

Dest: 0.0.0.0, NAT: Nir-NAT

Detach Elastic IP from VPC.

NAT Gateways → Nir-NAT → Remove → Delete

Elastic IPs → Stick → Release EIP → Release

Before Stop Instances

Remove volumes → Detach

Your VPC → Nir-VPC → Delete

Key Pair → Delete

⑥ Load Balancing

Create 2 instances (Linux)

Load Balancer → APP → Name → Create
TGr → Select 2 instance → include → Create
Back → Select TGr → Security Group (All) →
Zone (All) → Create (Copy LB(DNS))

DESKTOP

Volume Network
Subnet → any

① Launch Instance

② Volume → Create → Size = 5 →
Actions → Attach Instance →
Create x vdb

Connect
This PC → Manage → Tools → Computer
Management → Disk Management →
Disk 1 → Click Online again → Click
Initialize → Next box → Click →
New Simple Volume (New folder)

Next Instance →

Created Volume (Select) →
Actions → Create Snapshot →
Create

③ Snapshot

Create Snap (Select) → Actions
Create volume from Snap (Create)

④ Volume

Newly created (Attach) →
Instance (2) → x vdb (Create)

Run 2nd Instance

Repeat Step Online Only

Linux - 4 (run)
Windows - 5 (idle)

- ⑦ LB & Auto Scaling
 ⑧ Create VPC Instance
 ⑨ Load Balancing → Target Groups →
 Create → Choose Target → Instances
 Name: NiviTG1, Port: HTTP - 80, IPv4,
 VPC: default, Version: HTTP1,
 Health Check → Advanced → Healthy
 Threshold: 5, ^(default) **Next**
 Register Target **Create**
- ⑩ Load Balancer → Application →
 Name: NiviELB, Scheme: TCP, IPv4
 Network Mapping: VPC, Zone:
 Choose 26 Zones, Security Groups:
 default, Iw-5. Listeners & routing:
 NiviTG1 **Create**
- ⑪ Auto Scaling Group
 Create → Name: NiviAS, Launch
 Template → Create → Name: NiviTemp
 Application & OS Images: Ubuntu,
 Instance type: t2.micro (Free), KP: Nivi
 Network Settings → existing - Iw5,
 Advanced details → #!/bin/bash
 sudo apt-get update
 sudo apt-get install -y apache2
 sudo su
 \$ \$(hostname -f)* >/var/www/html/index.html
 echo "Welcome" > /var/www/html/index.html

VPC - IGW - T ELB
 AS (subnet) - LaunchTemp

ASG → Name
Target Group → Create → Put

NaviTG → Next [Create]

Load Balancer → Appli → Name:
New Security → Inbound rules: Anywhere → Create
LB → Security Groups → Selected Navi TG in Listener
[Create LB]

Instances under Launch Templates →
Create → Name: NaviTemp, OS Images:
Quick Start → Ubuntu, Type: Free,
KP Name: navikeyPair, Network Settings:
NaviSG

Advanced details

#!/bin/bash

Dudo apt-get update

Dudo apt-get install -y apache2

Echo "Hello \$(hostname -f)" > /var/www/html/index.html

[Create]

Auto Scaling Groups → Create →

Name: NaviASG

Launch template: NaviTemp → Next

Network: Availability Zone: Select all →

Next → Load balancing: Attach

existing → Attach → NaviTG →

Health Check → Turn on Elastic Load

Additional Settings → Enable → 15 min

Next → Group Size → Type: Unit

Capacity: 2

Scaling Min: 1 Max: 4

Automatic Scaling → Target Tracking

[Next] [Create ASG]

Copy Load Balancer DNS & paste URL



Select 1st instances → Connect
Select 2nd instances → Connect
Sudo Su (Type in 1st, 2nd)
top (check CPU utilization)

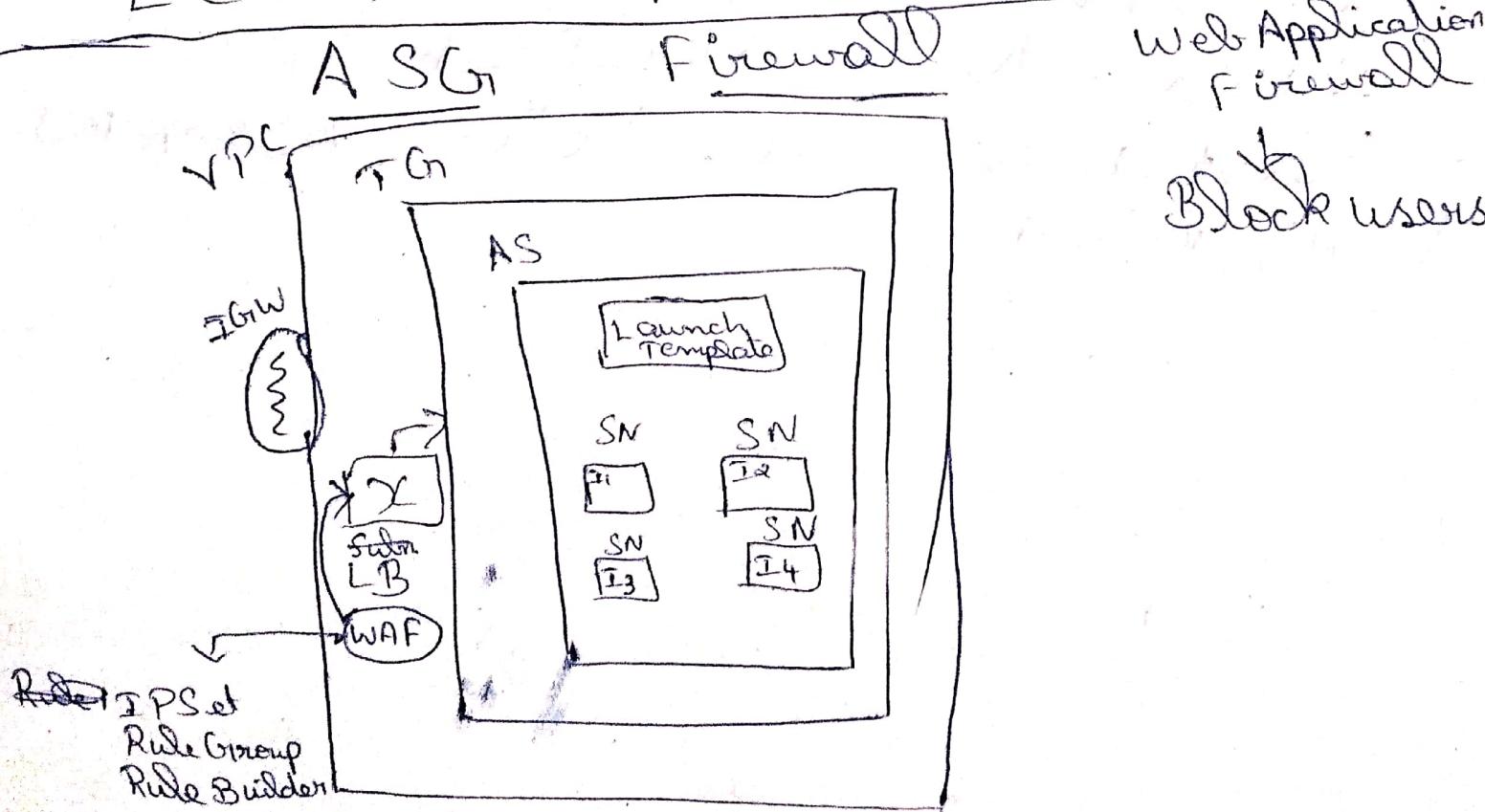
Q
yes > /dev/null & { Same as another instance
top (check CPU utilization)}

EC2 Home Page → Auto Scaling Group →
EC2 Instances

Select New ASG → Monitor → EC2 (check
CPU utilization) Refresh (it goes above 90% above)

Activity → History → 2 Instances

CPU 90% → Activity → New History (New instance created)
Decrease processid
EC2 → 3731 & kill 2717



ECD → Ubuntu ^{WAF} → Adv details (Linux)

[Create]

TG LB → Select 1st

LB → Select TG → Copy DNS → Paste

WAF & Shield → Create new ACL (Access Control List) → Check Region → Name: NiviACL

CloudWatch Metric Name: NiviACL →

Associated AWS → Add AWS resources →

Application LB → Select NiviLB (Add) →

Next → Add rules & rule groups ↓

Rules (Add rules) → Add my own
rules & rule group → IP Set → Region →

Name: NiviIPSet, IP address: ^{new} [Create]

210.210.254.67/32

Rule → Name: Nivi-Block, IP Set:

Select: NiviIPSet, Action: Block [Add]

Next

Create

Instance: 24/7, 1 Pro, 1 Pa

VPC

↓
1. Don't L. Pro +
2. Pro + AMP

Pro



IAM - Identity & Access Management

IAM → Users → Create → Name: demo
[Next] → Attach policies directly →
[Default (1207)] Permissions Policies: S3 →
Select: AmazonS3FullAccess → [Next]
[Create]

Click: → demo (in Dashboard),
Security Credentials; Copy:
Console Sign-in.
Enable access → Auto → enable →
Password created (copy).
Check Paste URL → Uname: demo → Pass