Day 1:

1. Login to Aws server and create the Ec2 instance with AMI id .

2. Steps to Generate the ssh-key file: refer the doc ssh-key.md file

3. download the ssh-keypair from aws and use the following command to login with out password.

ssh -i devops\_new1.pem centos@54.196.88.48

IN Linux every thing is command.

Linux is case sensitive.

--> To get the architure of OS:

$ uname -i

x86\_64

Note: 32-bit -i386/i586/i686 than it is 32-bit

---> to get the os details:

$ uname

Linux

---> to get the os vendor

cat /etc/\*release

$ cat /etc/\*release

CentOS Linux release 7.7.1908 (Core)

NAME="CentOS Linux"

VERSION="7 (Core)"

ID="centos"

ID\_LIKE="rhel fedora"

VERSION\_ID="7"

PRETTY\_NAME="CentOS Linux 7 (Core)"

ANSI\_COLOR="0;31"

CPE\_NAME="cpe:/o:centos:centos:7"

HOME\_URL="https://www.centos.org/"

BUG\_REPORT\_URL="https://bugs.centos.org/"

CENTOS\_MANTISBT\_PROJECT="CentOS-7"

CENTOS\_MANTISBT\_PROJECT\_VERSION="7"

REDHAT\_SUPPORT\_PRODUCT="centos"

REDHAT\_SUPPORT\_PRODUCT\_VERSION="7"

CentOS Linux release 7.7.1908 (Core)

CentOS Linux release 7.7.1908 (Core)

---> To check the CPU information:

$ cat /proc/cpuinfo

processor : 0

vendor\_id : GenuineIntel

cpu family : 6

model : 63

model name : Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz

stepping : 2

microcode : 0x43

cpu MHz : 2400.130

cache size : 30720 KB

physical id : 0

siblings : 1

core id : 0

cpu cores : 1

apicid : 0

initial apicid : 0

fpu : yes

fpu\_exception : yes

cpuid level : 13

wp : yes

flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse

sse2 ht syscall nx rdtscp lm constant\_tsc rep\_good nopl xtopology eagerfpu pni pclmulqdq ssse3 fma cx16 pcid sse4\_1 sse4\_2 x2apic movbe popcnt tsc\_deadline\_timer aes xsave avx f16c rdrand hypervisor lahf\_lm abm invpcid\_single fsgsbase bmi1 avx2 smep bmi2 erms invpcid xsaveopt

bogomips : 4800.05

clflush size : 64

cache\_alignment : 64

address sizes : 46 bits physical, 48 bits virtual

power management:

--> To check the memory information:

$ cat /proc/meminfo

MemTotal: 1013044 kB

MemFree: 842128 kB

MemAvailable: 814608 kB

Buffers: 4172 kB

Cached: 75240 kB

SwapCached: 0 kB

Active: 60248 kB

Inactive: 54928 kB

Active(anon): 35940 kB

Inactive(anon): 6632 kB

Active(file): 24308 kB

Inactive(file): 48296 kB

Unevictable: 0 kB

Mlocked: 0 kB

SwapTotal: 839676 kB

SwapFree: 839676 kB

Dirty: 0 kB

Writeback: 0 kB

AnonPages: 35800 kB

Mapped: 19668 kB

Shmem: 6808 kB

Slab: 27556 kB

SReclaimable: 15260 kB

SUnreclaim: 12296 kB

KernelStack: 1856 kB

PageTables: 3168 kB

NFS\_Unstable: 0 kB

Bounce: 0 kB

WritebackTmp: 0 kB

CommitLimit: 1346196 kB

Committed\_AS: 227208 kB

VmallocTotal: 34359738367 kB

VmallocUsed: 10944 kB

VmallocChunk: 34359722492 kB

HardwareCorrupted: 0 kB

AnonHugePages: 8192 kB

CmaTotal: 0 kB

CmaFree: 0 kB

HugePages\_Total: 0

HugePages\_Free: 0

HugePages\_Rsvd: 0

HugePages\_Surp: 0

Hugepagesize: 2048 kB

DirectMap4k: 47104 kB

DirectMap2M: 1001472 kB

$ free -m

total used free shared buff/cache available

Mem: 989 74 822 6 92 795

Swap: 819 0 819

--> To view the disk information.

$ sudo fdisk -l

Disk /dev/xvda: 10.7 GB, 10737418240 bytes, 20971520 sectors

Units = sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk label type: dos

Disk identifier: 0x000a2a6d

Device Boot Start End Blocks Id System

/dev/xvda1 \* 2048 1026047 512000 83 Linux

/dev/xvda2 1026048 16777215 7875584 8e Linux LVM

Disk /dev/mapper/centos-root: 7159 MB, 7159676928 bytes, 13983744 sectors

Units = sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/centos-swap: 859 MB, 859832320 bytes, 1679360 sectors

Units = sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

[centos@ip-172-31-43-234 ~]$ $ free -m

-bash: $: command not found

[centos@ip-172-31-43-234 ~]$ total used free shared buff/cache available

-bash: total: command not found

[centos@ip-172-31-43-234 ~]$ Mem: 989 74 822 6 92 795

-bash: Mem:: command not found

[centos@ip-172-31-43-234 ~]$ Mem: 989 74 822 6 92 795

[centos@ip-172-31-43-234 ~]$ clar

-bash: clar: command not found

[centos@ip-172-31-43-234 ~]$ clear

[centos@ip-172-31-43-234 ~]$ fdisk -l

fdisk: cannot open /dev/xvda: Permission denied

fdisk: cannot open /dev/mapper/centos-root: Permission denied

fdisk: cannot open /dev/mapper/centos-swap: Permission denied

[centos@ip-172-31-43-234 ~]$ sudo fdisk -l

Disk /dev/xvda: 10.7 GB, 10737418240 bytes, 20971520 sectors

Units = sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk label type: dos

Disk identifier: 0x000a2a6d

Device Boot Start End Blocks Id System

/dev/xvda1 \* 2048 1026047 512000 83 Linux

/dev/xvda2 1026048 16777215 7875584 8e Linux LVM

Disk /dev/mapper/centos-root: 7159 MB, 7159676928 bytes, 13983744 sectors

Units = sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/centos-swap: 859 MB, 859832320 bytes, 1679360 sectors

Units = sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

##################################################################

Day 2:

Create a new EC2 server from AWS.

ls: list of files in currenty directoy

# ls

anaconda-ks.cfg core.16461 Desktop Documents etc\_bkp.gzip Music Public Videos

clientinstall.sh desktop desktop.pub Downloads initial-setup-ks.cfg Pictures Templates

--> in Linux any folder starting with . will be treated as hidden directory.

--> to create hidden folder/file

--> mkidir .fodername/filename

--> to list hidden files:

ls -lA or ls -a

---> to escape the some feature of command use \commmnad

Ex: #\ls -lA

note: by prefixing the "\" dont use aliasis and functions, use only builtin commands

----------------------------------------------------------------------------------------

--> Just observe the output:

# which ls

alias ls='ls --color=auto'

/usr/bin/ls

[root@desktop ~]# /usr/bin/ls

anaconda-ks.cfg core.16461 Desktop Documents etc\_bkp.gzip Music Public Videos

clientinstall.sh desktop desktop.pub Downloads initial-setup-ks.cfg Pictures Templates

[root@desktop ~]# /usr/bin/ls --color=auto

anaconda-ks.cfg core.16461 Desktop Documents etc\_bkp.gzip Music Public Videos

clientinstall.sh desktop desktop.pub Downloads initial-setup-ks.cfg Pictures Templates

-------------------

-> to set some alias for a command. hear i am changing ls commmand alisa command of date

[root@desktop ~]# alias ls='date'

[root@desktop ~]# ls

Sat Dec 7 12:21:07 IST 2019

[root@desktop ~]# date

Sat Dec 7 12:21:10 IST 2019

[root@desktop ~]# unalias ls

[root@desktop ~]# ls

anaconda-ks.cfg core.16461 Desktop Documents etc\_bkp.gzip Music Public Videos

clientinstall.sh desktop desktop.pub Downloads initial-setup-ks.cfg Pictures Templates

-------------

alias::::

[root@desktop ~]# alias ls='date'

[root@desktop ~]# ls

Sat Dec 7 12:21:07 IST 2019

[root@desktop ~]# date

Sat Dec 7 12:21:10 IST 2019

[root@desktop ~]# un

unalias uncompress unicode\_stop unix-lpr.sh unlz4 unshare unzip

uname unexpand uniq unix\_update unpack200 until unzipsfx

unbound-anchor unicode\_start unix\_chkpwd unlink unset unxz

[root@desktop ~]# unalias ls

[root@desktop ~]# ls

anaconda-ks.cfg core.16461 Desktop Documents etc\_bkp.gzip Music Public Videos

clientinstall.sh desktop desktop.pub Downloads initial-setup-ks.cfg Pictures Templates

---------------------

Links:::

# ln -s mainfile.txt mainfile\_s.txt

[root@desktop ~]# ls -ltr main\*

-rw-r--r-- 1 root root 10 Dec 7 12:38 mainfile.txt

lrwxrwxrwx 1 root root 12 Dec 7 12:39 mainfile\_s.txt -> mainfile.txt

[root@desktop ~]# ls -li main\*

34889556 lrwxrwxrwx 1 root root 12 Dec 7 12:39 mainfile\_s.txt -> mainfile.txt

34889696 -rw-r--r-- 1 root root 10 Dec 7 12:38 mainfile.txt

[root@desktop ~]# ln mainfile.txt mainfile\_h.txt

[root@desktop ~]# ls -li main\*

34889696 -rw-r--r-- 2 root root 10 Dec 7 12:38 mainfile\_h.txt

34889556 lrwxrwxrwx 1 root root 12 Dec 7 12:39 mainfile\_s.txt -> mainfile.txt

34889696 -rw-r--r-- 2 root root 10 Dec 7 12:38 mainfile.txt

[root@desktop ~]# unlink mainfile\_s.txt

----------------

Copy a file or folder:::

1. cp

2. rsync

both the commands are very similar to copy the files.

# cp etc\_bkp.gzip Desktop/etc\_bkp

[root@desktop ~]# rsync etc\_bkp.gzip Desktop/etc\_bkp1

[root@desktop ~]# cd Desktop/

[root@desktop Desktop]# ls

CentOS-7-x86\_64-DVD-1810.iso etc\_bkp etc\_bkp1

----------------------------

File overwrite:

# cp etc\_bkp1 etc\_bkp

cp: overwrite ‘etc\_bkp’? y

[root@desktop Desktop]# cp etc\_bkp etc\_bkp1

cp: overwrite ‘etc\_bkp1’? y

[root@desktop Desktop]# cp etc\_bkp etc\_bkp1 -y

cp: invalid option -- 'y'

Try 'cp --help' for more information.

[root@desktop Desktop]# cp -f etc\_bkp1 etc\_bkp

cp: overwrite ‘etc\_bkp’? ╚

[root@desktop Desktop]# \cp -f etc\_bkp1 etc\_bkp

[root@desktop Desktop]# alias cp

alias cp='cp -i'

# in linux some of the commands are aliased as below.

# cp etc\_bkp1 etc\_bkp

cp: overwrite ‘etc\_bkp’? y

[root@desktop Desktop]# cp etc\_bkp etc\_bkp1

cp: overwrite ‘etc\_bkp1’? y

[root@desktop Desktop]# cp etc\_bkp etc\_bkp1 -y

cp: invalid option -- 'y'

Try 'cp --help' for more information.

[root@desktop Desktop]# cp -f etc\_bkp1 etc\_bkp

cp: overwrite ‘etc\_bkp’? ╚

[root@desktop Desktop]# \cp -f etc\_bkp1 etc\_bkp

[root@desktop Desktop]# alias cp

alias cp='cp -i'

---------------------------------------

--> To copy a file with the file permissions...

cp -p sourcefile destfile

-------------------------------

to search a text

grep manas\* -R -l

-------------------------------

Day 4..

Archiving:

-----

Day 5:

package management:

yum list available

-->to list all the services which are active

# systemctl list-units -t service | grep httpd

httpd.service loaded active running The Apache HTTP Server

-->to list all the services which are active / inactive

# systemctl list-units -t service -a | grep http

httpd.service loaded inactive dead The Apache HTTP Server

few times it willnot show correclty. better use below command.

# systemctl list-unit-files --all | grep httpd

httpd.service disabled

Note: The main difference between Web server and application server is that web server is meant to serve static pages e.g. HTML and CSS, while Application Server is responsible for generating dynamic content by executing server side code e.g. JSP, Servlet or EJB

By default webserver is installed in application server. Please refer the snap for the further details.

-------------

To export the display in Gui in linux.

we need to install one utility xming --> launch xming. set the display port as 7.0

# export DISPLAY=192.168.1.23:7.0 ---> ipaddress of windows machine with the port 7.0

ln

[root@desktop ~]# firewall-config ---> now gui working.

NOte: this step will not work in AWS.

if we are connecting from linux machine gui, you can directly use the ssh username@ipaddress -X

----------------------------

Day 5:

to know about the port no.

vim /etc/services

--> now we are changing the Listning portno .

1. change the Listen Port 80 t0 8888

vim /etc/httpd/conf/httpd.conf

2. now restart the httpd service

# systemctl restart httpd

3. we need to enable the firewall of 8888

# firewall-cmd --permanent --add-port=8888/tcp

success

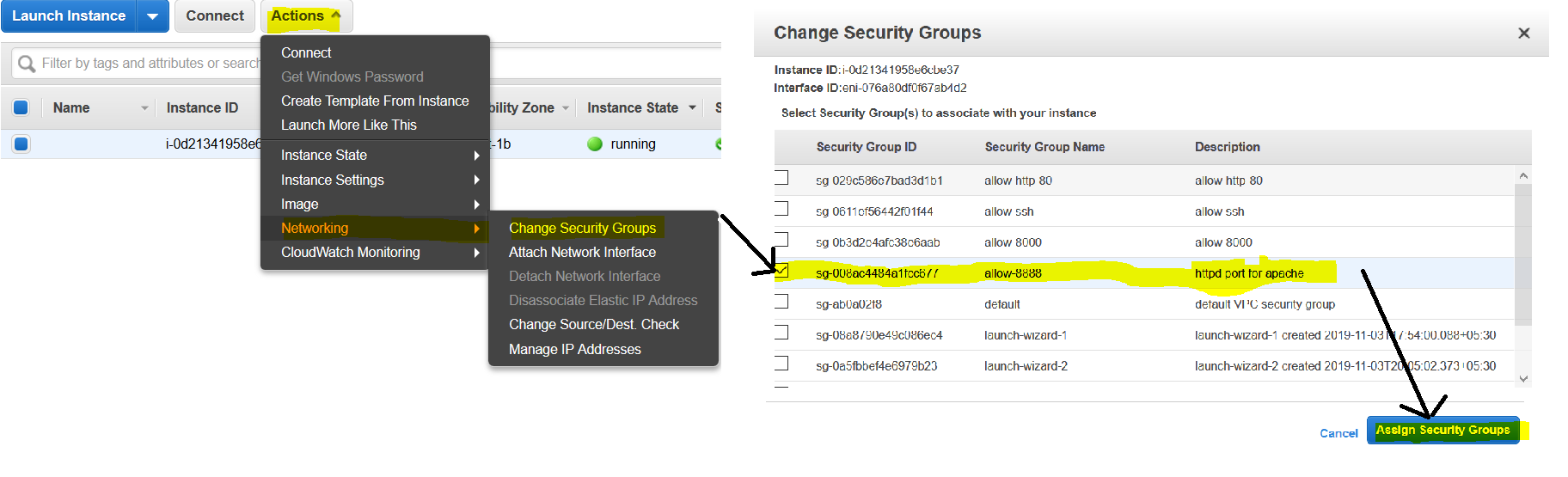
4. # firewall-cmd --reload

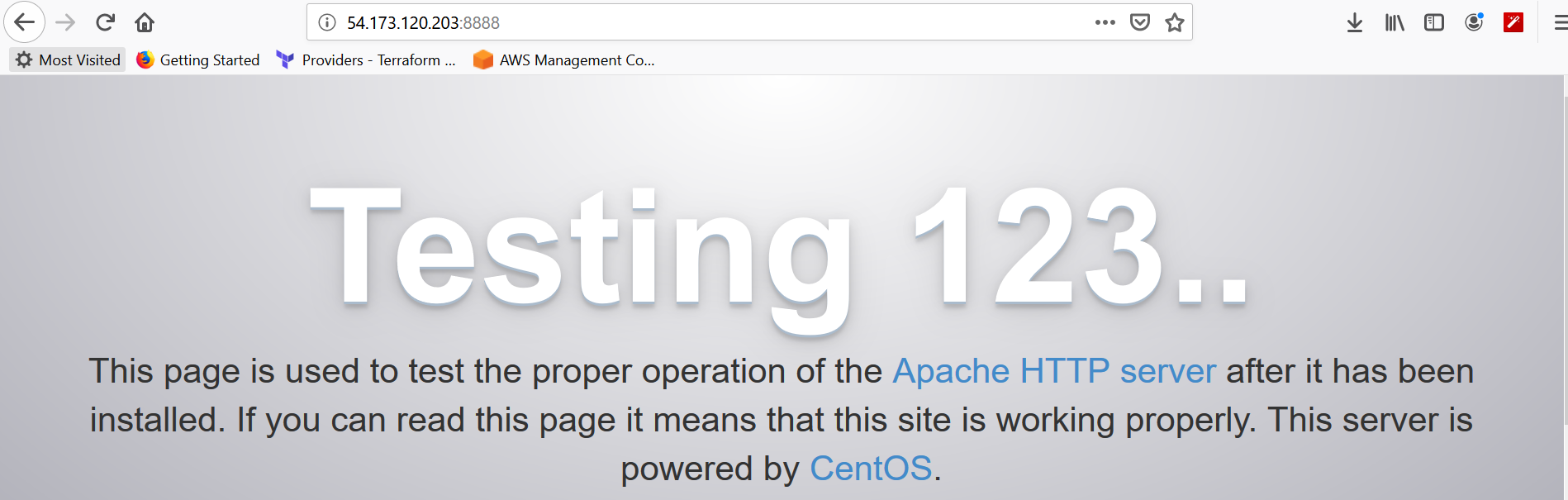
success

5. open any browser and type hostname:8888 --> it will open default apache page

Note: this is manual configuration in legacy machines.

for in the Aws cloud we need to security groups as below.



b

![img](apache.png)

--🡪 create a template 🡪 create instance from templates.

-------------🡪

Linux file permissions:

[root@ip-172-31-46-86 ~]# touch sample

[root@ip-172-31-46-86 ~]# ls -l

total 0

-rw-r--r-- 1 root root 0 Dec 8 05:01 sample

[root@ip-172-31-46-86 ~]# chown u1 sample

[root@ip-172-31-46-86 ~]# ls -l

total 0

-rw-r--r-- 1 u1 root 0 Dec 8 05:01 sample

[root@ip-172-31-46-86 ~]# chgrp u1 sample

[root@ip-172-31-46-86 ~]# ls -l

total 0

-rw-r--r-- 1 u1 u1 0 Dec 8 05:01 sample

[root@ip-172-31-46-86 ~]# chown root:root sample

[root@ip-172-31-46-86 ~]# ls -l

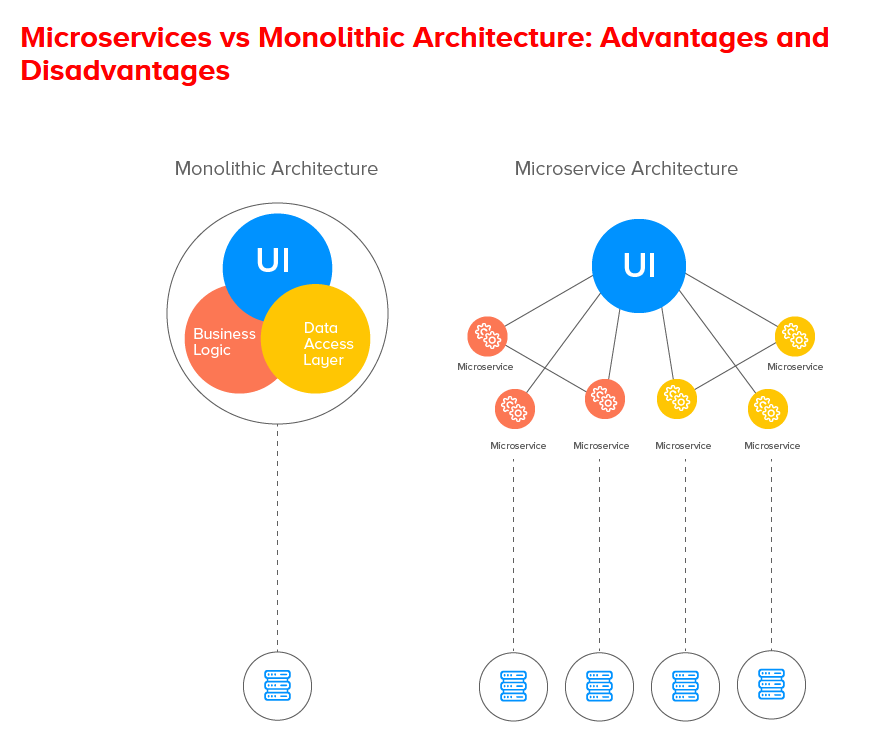
total 0

-rw-r--r-- 1 root root 0 Dec 8 05:01 sample

Note: By using above method, we can able to change the owner and group in a single command.

Day 6:

Monolithic vs microservices architure.



Microservice –New

Monolithic -Old

The problem with the monolithic application is some thing is not running your complete application is down.

Back in 2008, when Netflix was still operating as a monolith, a single missing semi semicolon brought down the entire Netflix website for several hours. Monoliths tend to become spaghetti code with various components linked together and tightly coupled together. As a monolith, when something was broken in the code, all Netflix engineers had to be alerted to check and see if it was their code that had caused the problem. While it is still possible to have outages with a microservices architecture due to code errors and other issues, outages can be minimized since the platform is divided into separate services that are loosely coupled. A well-designed microservices architecture allows for better availability.

Reference : <https://smartbear.com/blog/develop/why-you-cant-talk-about-microservices-without-ment/>

Apache Vs NGINX🡪

Deploying the Webserver with NGINX:

1. **Login to AWS**
2. **Launch the instance with template**
3. **Check the yum package of nginx , but it is 1.16**
4. **Download latest at** [**http://nginx.org/download/nginx-1.17.6.zip**](http://nginx.org/download/nginx-1.17.6.zip)
5. **# wget** [**http://nginx.org/download/nginx-1.17.6.tar.gz**](http://nginx.org/download/nginx-1.17.6.tar.gz)
6. **# tar -xvzf nginx**

**7.# cd nginx-1.17.6**

[root@ip-172-31-42-63 nginx-1.17.6]# ls

auto CHANGES CHANGES.ru conf configure contrib html LICENSE man README src

**8.[root@ip-172-31-42-63 nginx-1.17.6]# ./configure**

checking for OS

+ Linux 3.10.0-1062.1.1.el7.x86\_64 x86\_64

checking for C compiler ... not found

./configure: error: C compiler cc is not found

After nginx manual installation we need to in site with dns

Ref: freenom and dns checker org

**9.# yum install gcc -y**

**10.** **# ./configure**

**./configure: error: the HTTP rewrite module requires the PCRE library.**

**You can either disable the module by using --without-http\_rewrite\_module**

**option, or install the PCRE library into the system, or build the PCRE library**

**statically from the source with nginx by using --with-pcre=<path> option.**

**11. yum install pcre\* -y**

**12. ./configure**

**./configure: error: the HTTP gzip module requires the zlib library.**

**You can either disable the module by using --without-http\_gzip\_module**

**option, or install the zlib library into the system, or build the zlib library**

**statically from the source with nginx by using --with-zlib=<path> option.**

**13. # yum install zlib\* -y**

**14. ./configure**

**This time we don’t get any errors. In the next line to confirm the exit status.**

**15.** **# echo $?**

**0**

**16. Now make file has been created.**

**17. make**

**18. check the exit staus.**

**18. make install**

**20. echo $?**

**0**

**Now Nginx successfully installed.**

**To start the nginx server.**

**Cd /usr/local/nginx/**

**./nginx**

**Check the firewall opened or not.**

**# netstat -lntp**

**Active Internet connections (only servers)**

**Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name**

**tcp 0 0 0.0.0.0:80 0.0.0.0:\* LISTEN 8502/nginx: master**

**tcp 0 0 0.0.0.0:22 0.0.0.0:\* LISTEN 1027/sshd**

**tcp6 0 0 :::22 :::\* LISTEN 1027/sshd**

**Or**

**[root@ip-172-31-42-63 sbin]# ss -tulpn**

**Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port**

**udp UNCONN 0 0 \*:68 \*:\***

**users:(("dhclient",pid=764,fd=6))**

**tcp LISTEN 0 128 \*:80 \*:\***

**users:(("nginx",pid=8503,fd=6),("nginx",pid=8502,fd=6))**

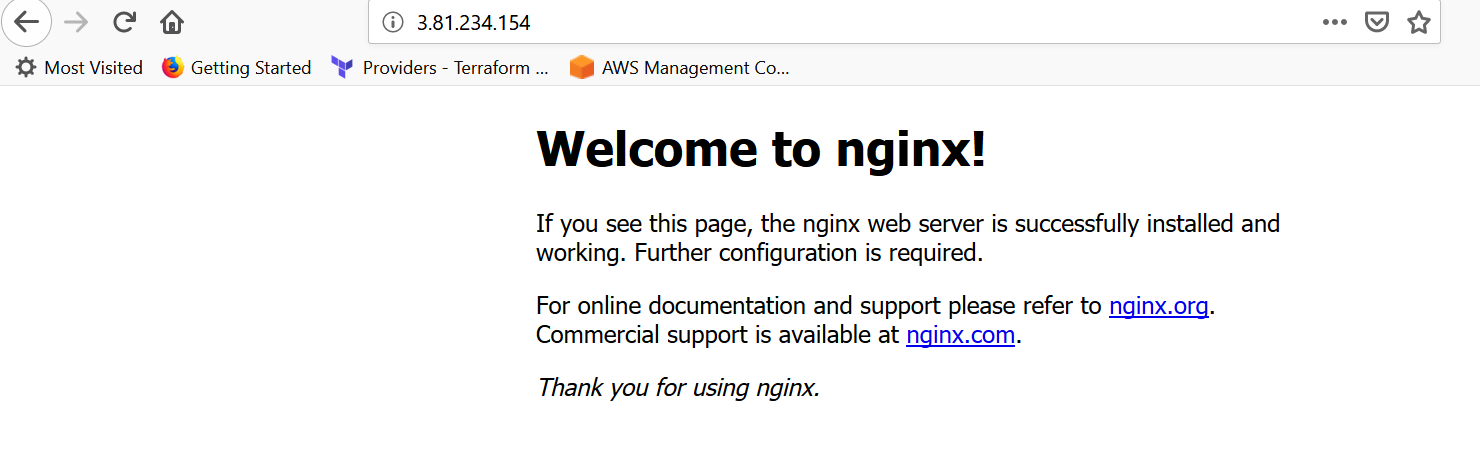
**tcp LISTEN 0 128 \*:22 \*:\***

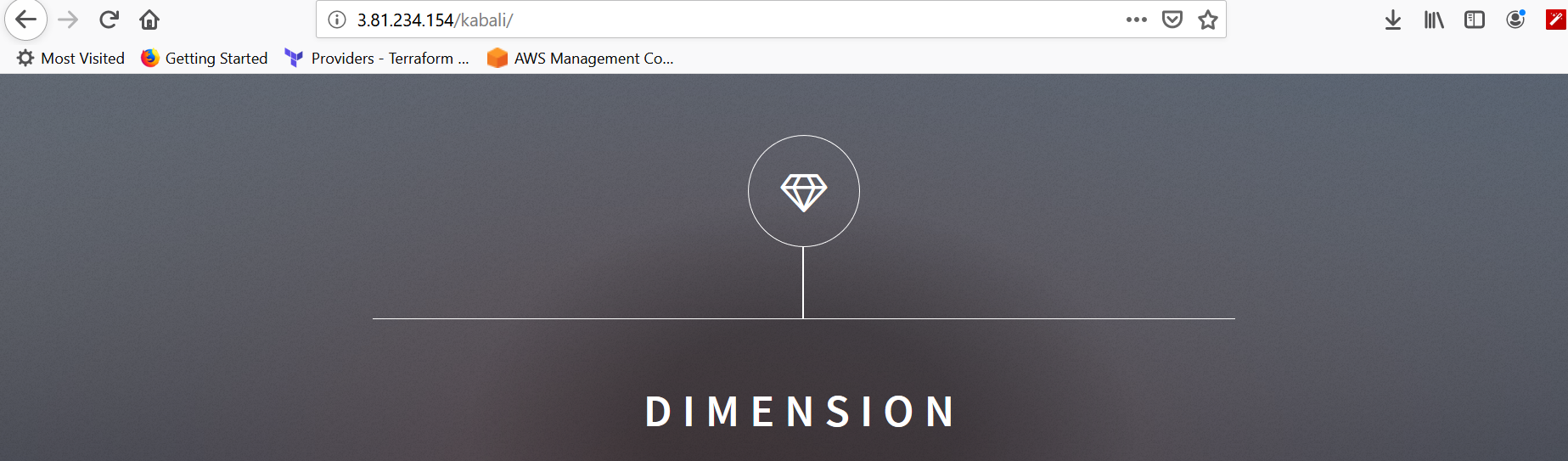
**users:(("sshd",pid=1027,fd=3))**

**tcp LISTEN 0 128 [::]:22 [::]:\***

**users:(("sshd",pid=1027,fd=4))**

**Now the index page is browsable with ip address.**





* **Go to services in AWS 🡪 route 53 🡪 DNS management 🡪 get started🡪**
* **Create Hosted Zone 🡪 Create Hosted Zone --.create**



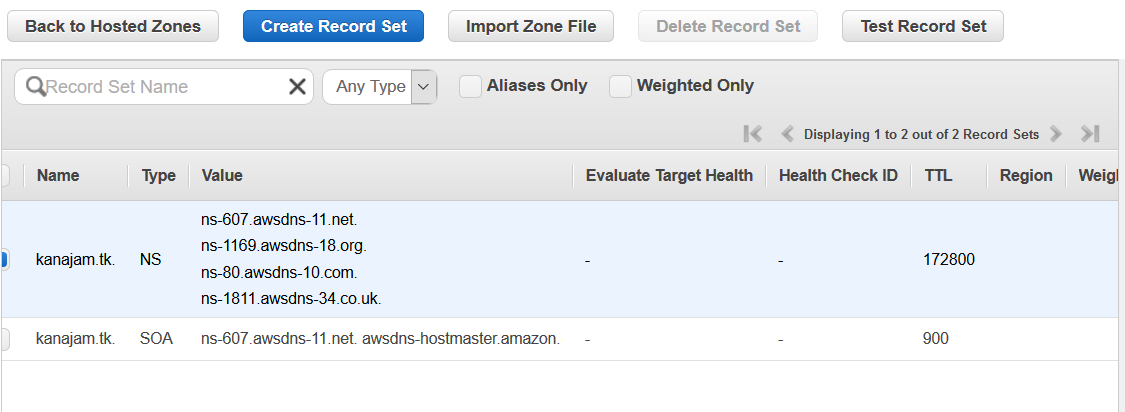
**Copy the name server record values.**

**ns-607.awsdns-11.net.**

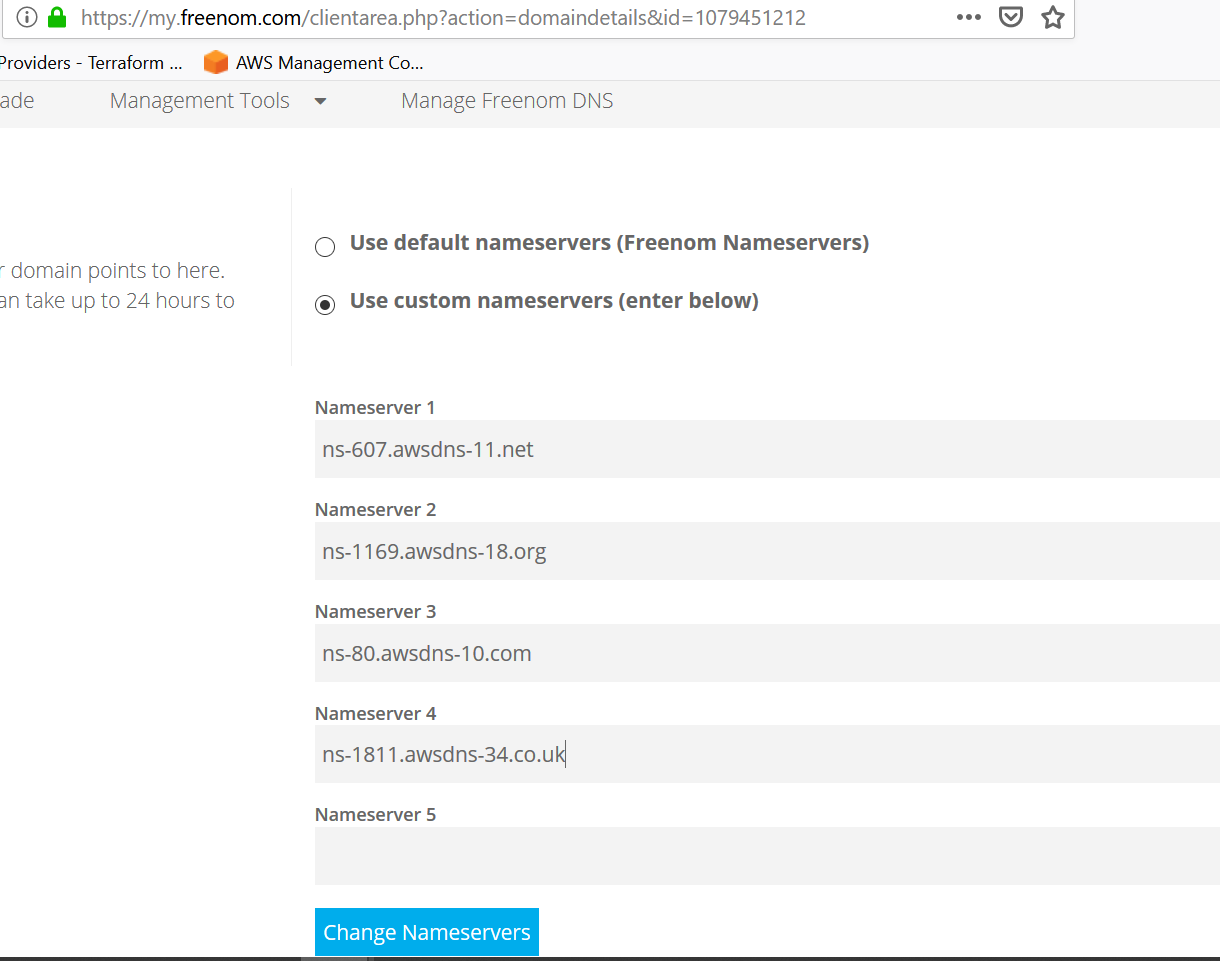
**ns-1169.awsdns-18.org.**

**ns-80.awsdns-10.com.**

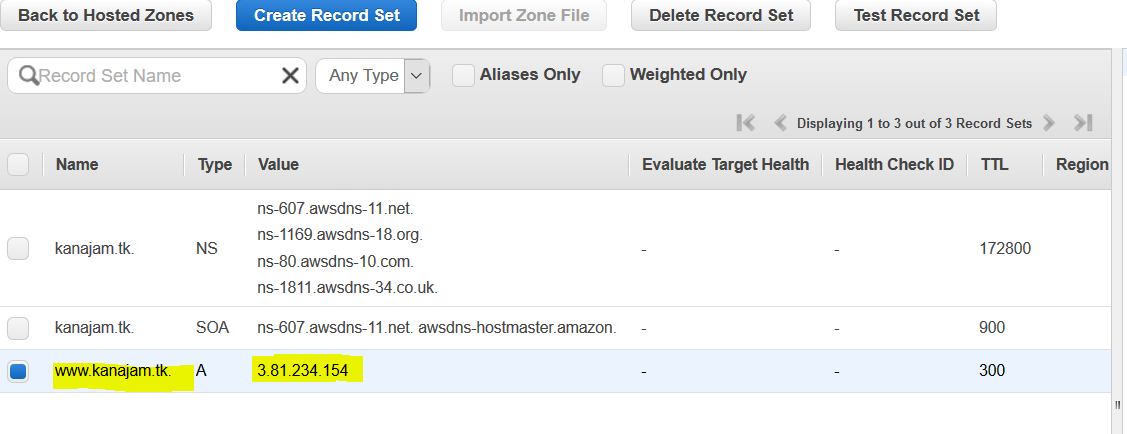
**ns-1811.awsdns-34.co.uk.**

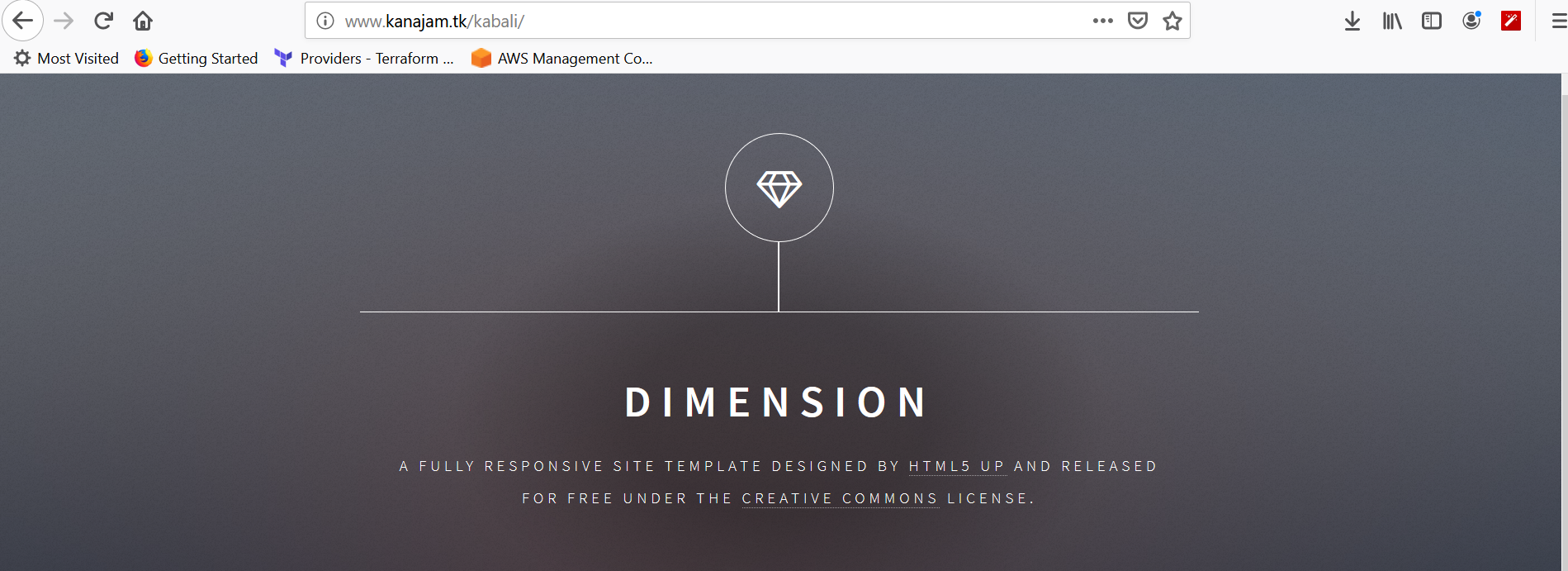


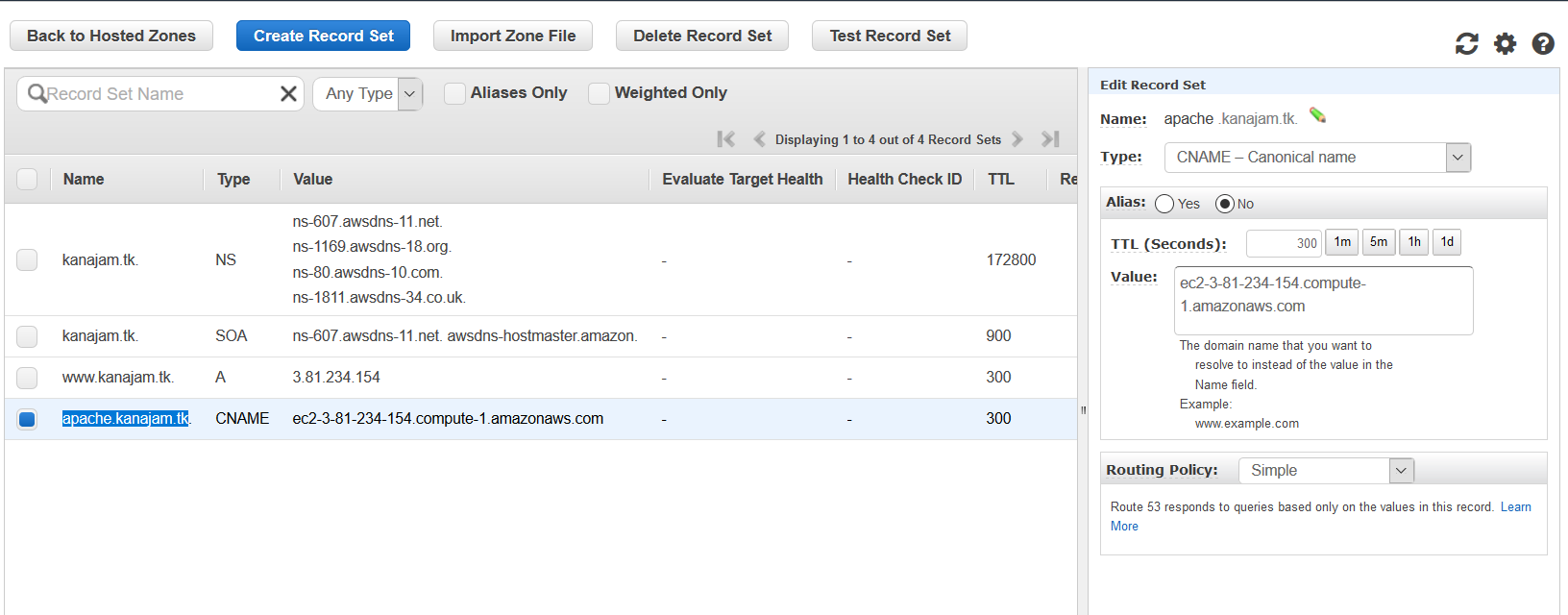
**Login to freenom site or some domain site provide the name server records.**



**Create record set 🡪**

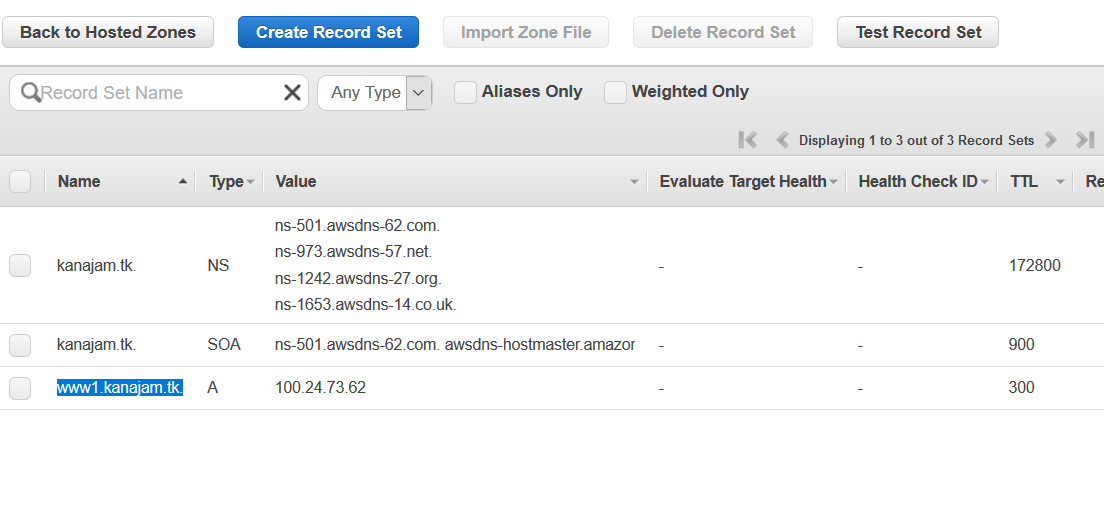


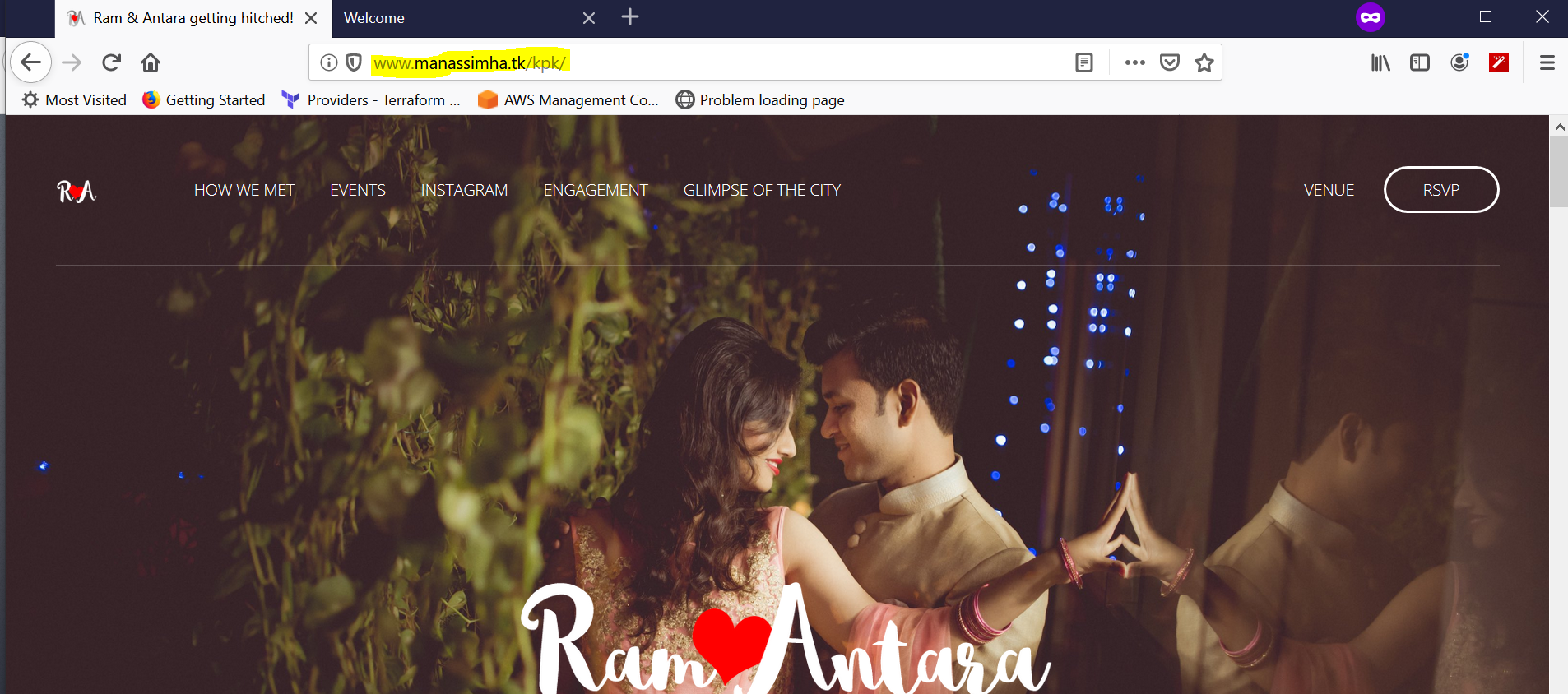


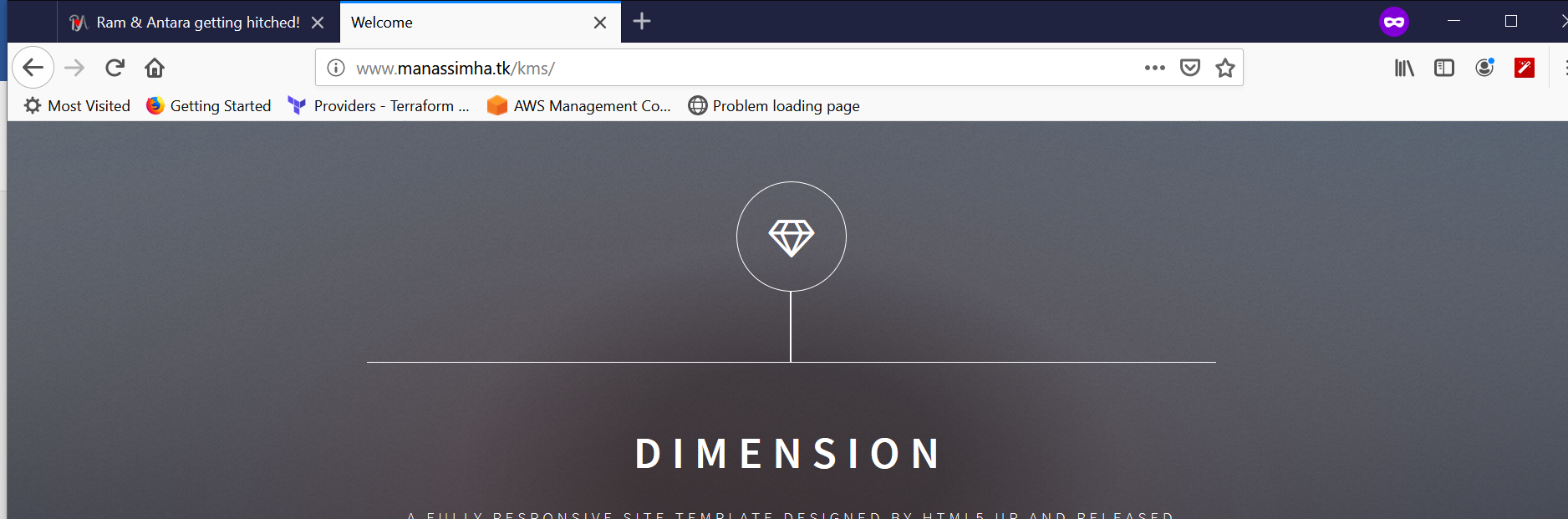


**If you don’t have name then use A record and provide some name with ip**

**If already having a name , we can use cnmae .**

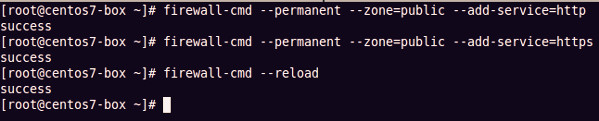






In the above example, the web pages <http://www.manassimha.tk/kms/> and <http://www.manassimha.tk/kpk/> need to changes to name based web hosting.

**Step 4 – Open port 80 and 443 using firewall-cmd**

You must open and enable port 80 and 443 using the firewall-cmd command:  
$ sudo firewall-cmd --permanent --zone=public --add-service=http  
$ sudo firewall-cmd --permanent --zone=public --add-service=https  
$ sudo firewall-cmd --reload  
[](https://www.cyberciti.biz/media/new/faq/2018/01/RHEL7-CentOS7-allow-HTTP-and-HTTPS-traffic-with-firewall-cmd.jpg)

**Step 5 – Test it**

Verify that port 80 or 443 opened using ss command:  
$ sudo ss -tulpn

-🡪 check video 7 for nginx and apache.