- Deploy the VPC Cloudformation Template that will create Public and Private Subnet
- Launch 1 EC2 in Public Subnet and 1 EC2 in Private Subnet with Centos AMI with atleast 20GB as Root EBS
- make sure all commands are execute with root user in linux
- Copy the .pem key to public instance

```
scp -i aws-vpc-key.pem aws-vpc-key.pem centos@<EC@_PUBLIC_IP>:~/
```

- Install httpd server on public EC2 that will host the repo files:

```
sudo yum install tree httpd -y
```

- Connect to Private EC2 from Public EC2.

```
sudo yum install httpd
```

- The above command will not work since there is no internet connection to Centos Base Repo as this EC2 machine is not having NAT Gateway also
- On Public EC2 instance
- One helpful tool is the `createrepo` software package. This software bundles several `.rpm` files together into a `repomd` repository. -Install the software by entering the following:

```
sudo yum install createrepo yum-utils -y
```

- To check history of the packages

```
yum history list
```

- To check more details of yum transaction 3

```
yum history info 3
```

- Create a directory to store the repositories

```
sudo mkdir -p /var/www/html/repos/{base,centosplus,extras,updates}

tree /var/www/html/
```

- Check Output of below command:

```
yum repolist
```

- Synchronize HTTP repositories

```
sudo reposync -g -l -d -m --repoid=base --newest-only --download-metadata --
download_path=/var/www/html/repos/
sudo reposync -g -l -d -m --repoid=centosplus --newest-only --download-metadata --
download_path=/var/www/html/repos/
sudo reposync -g -l -d -m --repoid=extras --newest-only --download-metadata --
download_path=/var/www/html/repos/
sudo reposync -g -l -d -m --repoid=updates --newest-only --download-metadata --
download_path=/var/www/html/repos/
```

The Parameters in the above command are as below:

- `--repoid` – specifies the repository ID.
- `--newest-only` – tell reposync to only pull the latest version of each package in the repos.
- `--download-metadata` – enables downloading all the non-default metadata.
- `--download_path` – specifies the path to download packages.
- `-g` – enables removing of packages that fail GPG signature checking after downloading.
- `-l` – enables yum plugin support.
- `-d` – enables deleting of local packages no longer present in repository.
- `-m` – enables downloading of comps.xml files.
- Create a new repodata for the local repositories

```
sudo createrepo /var/www/html
```

> The above command will take around 5 minutes to execute

- On Private EC2 instance

- move the .repo extension files to /tmp directory, so that yum will not search repo on the internet mirrorlist.

```
mv /etc/yum.repos.d/*.repo /tmp/
```

- Use below command to create a `.repo` that will point to another EC2 that has all the packages installed.

```
cat > /etc/yum.repos.d/remote-ec2.repo <<EOF
[remote-ec2]
name=RHEL Apache
baseurl=http://Private-Ip-of-Public-EC2
enabled=1
gpgcheck=0
EOF
```

- Below commands should be able to list the packages that are downloaded from previous step `yum repolist`
- Make sure httpd service is running on Public EC2

```
systemctl start httpd
systemctl enable httpd
```

- Some more YUM commands to execute
- list all available packages `yum list available`
- to list all installed packages `yum list installed`
- list installed and available packages `yum list all`
- list installed and available kernel packages `yum list kernel`
- get info related to a package `yum info httpd`
- get the dependencies of a particular package `yum deplist httpd`
- search whether a particular package is available `yum search httpd`
- delete packages saved in cache `yum clean packages`
- clean out all packages and meta data from cache `yum clean all`
- install/update packages using yum

```
yum install httpd
yum update httpd
```

- if you have an .rpm file locally and you want to install package locally using yum `yum localinstall abc-1-1.i686.rpm`