Steps to create server on AWS and run Nginx with custom static file:

1. Create IAM service under AWS Management Console: This step is used to create IAM group, user, roles, policies and other access configurations

NOTE: Steps a, b and c are not part of server creating steps, but it is mandatory for creating an applications in org and allow/limit users

1. Make sure to set up security policies first by managing account settings under “Security Status”. And save all the imp credentials like Multi-factor Authentication, Access keys, Cloud key pairs (will be used in later steps) and other certificates.
2. Create user group for your desired task/app\_code.
3. Add users to the above created group who can be able to access its associated resources.
4. Create IAM role for instance creation on AWS management console (This is mandatory step for server creation performed in later steps).
5. Create EC2 instance on AWS Console: This step is used to create instance with required instance base image, type, storage, group and other settings for creation of server.
6. Launch instance from console.
7. Select base image for instance machine.
8. Select type.
9. Configure details like network, subnet, IAM role (refer step 1.c) or you can leave it as default.
10. Add storage, tag and security group if needed else just hit “Review and Launch”.
11. When you will launch the instance, pop-up option will ask to enter ec2 cloud key pair. This option usually display default key pair, if you have one created already (refer step 1.a). Else you need to create one first and follow step 1 to get instance created.
12. Set inbound rules for EC2 instances (under AWS management console): This step is used to specify the type of traffic our EC2 instance can be managed to handle for example: http, https or SSH and limit them as per required settings.
13. Edit inbound rules
14. Configure inbound rules with required network traffic details that your instances are expected. For example: Only certain IP can access EC2 instance via SSH, then set the rule under SSH type with custom IP option and add IP range.
15. Launch and install nginx on created instance.
16. Connect instance as browser-based SSH connection (We can also try SSH into instance).
17. Once you connect as ec2-user, you can see console of ec2 instance machine.
18. Set new password for root user > sudo password root
19. Try updating yum > sudo yum update
20. Install nginx > sudo yum install nginx
21. Start & Enable nginx service > sudo systemctl start nginx / sudo systemctl enable nginx
22. Check nginx status > sudo systemctl status nginx
23. Check on browser if nginx is up and running > curl localhost / accessing public DNS in browser
24. Replace nginx homepage with custom static file: Usually when nginx service is launched on any machine it will display custom index.html content on browser. This is configured in /etc/nginx/nginx.conf file in linux machines. We can tweak the file location in config file and make custom content displayed on nginx homepage.
25. Create version.txt file containing custom data.
26. Locate nginx config file: In this example, as we are using linux machine this file can be seen under /etc/nginx/nginx.conf
27. Replace server section with

server {

root /;

location / {

try\_files $uri /nginx/repos/version.txt;

}

1. The above tweak in server file location will display content of file “version.txt” at nginx homepage now as we swapped the content with index.html
2. Restart nginx service > sudo systemctl restart nginx
3. Check nginx service status > sudo systemctl status nginx
4. Check content of browser/ local host: curl localhost / accessing public DNS in browser