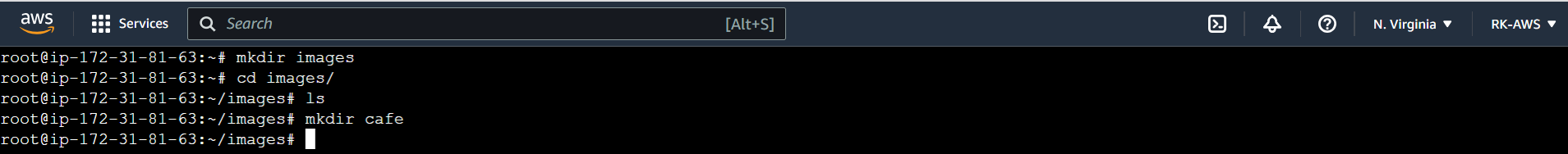
Containarise-a-webApp

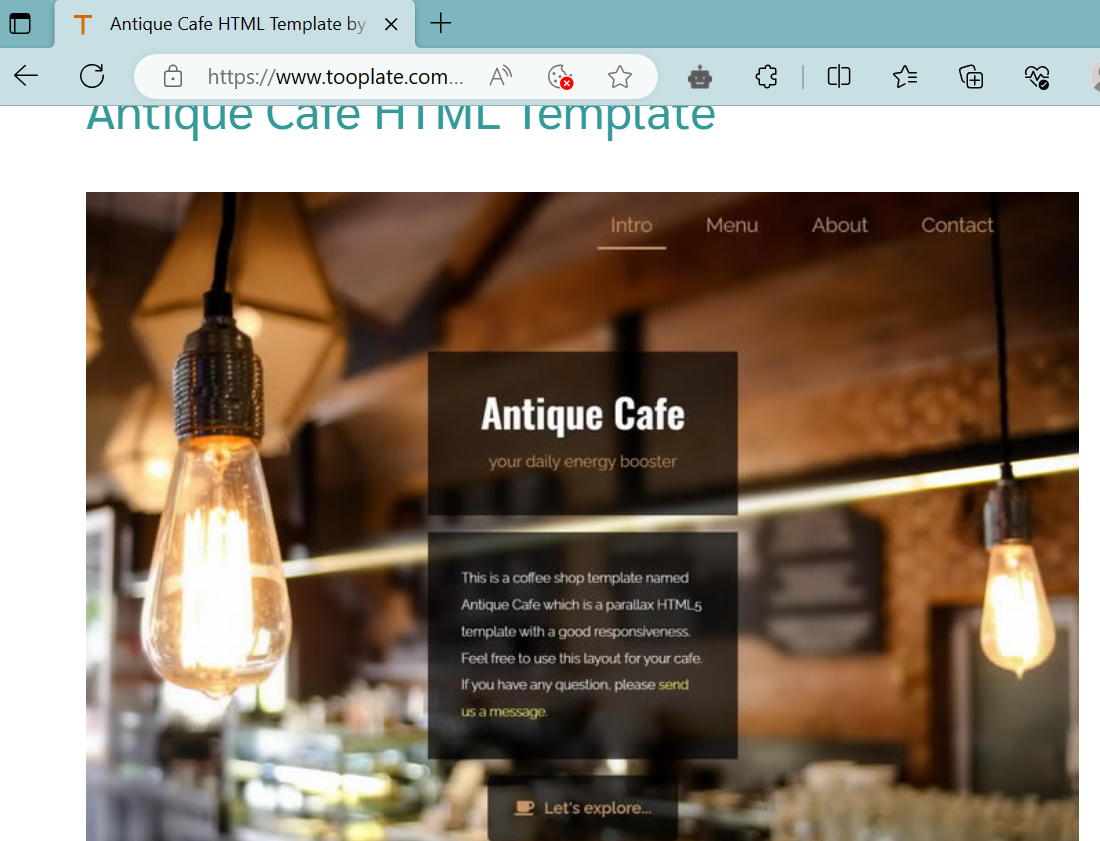
Pre-requisite : Docker Engine installed on AWS EC2

Steps :

1. Create a directory

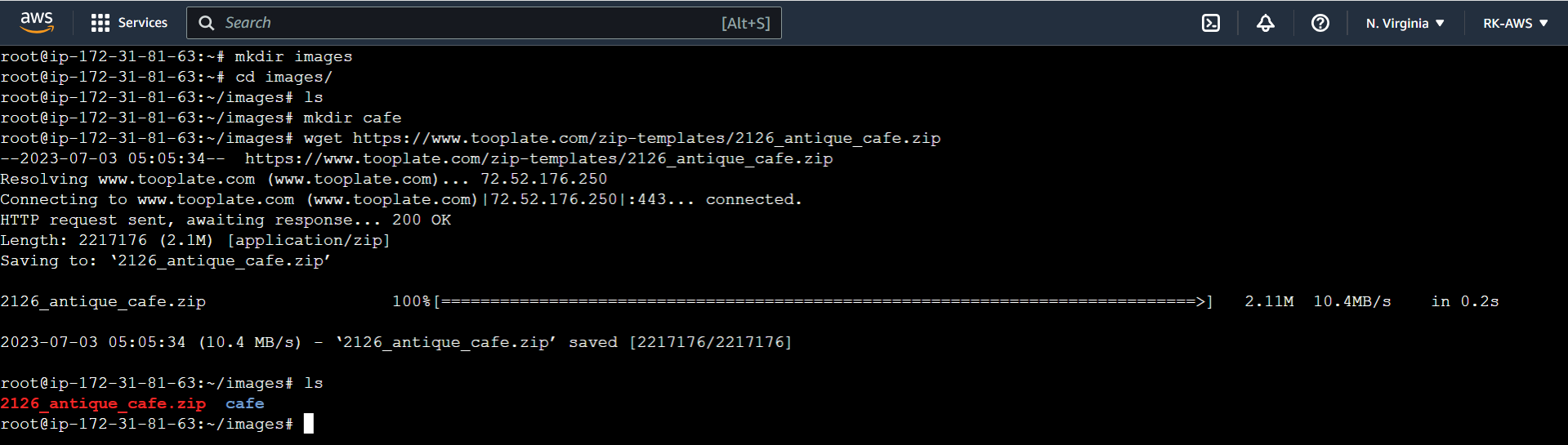


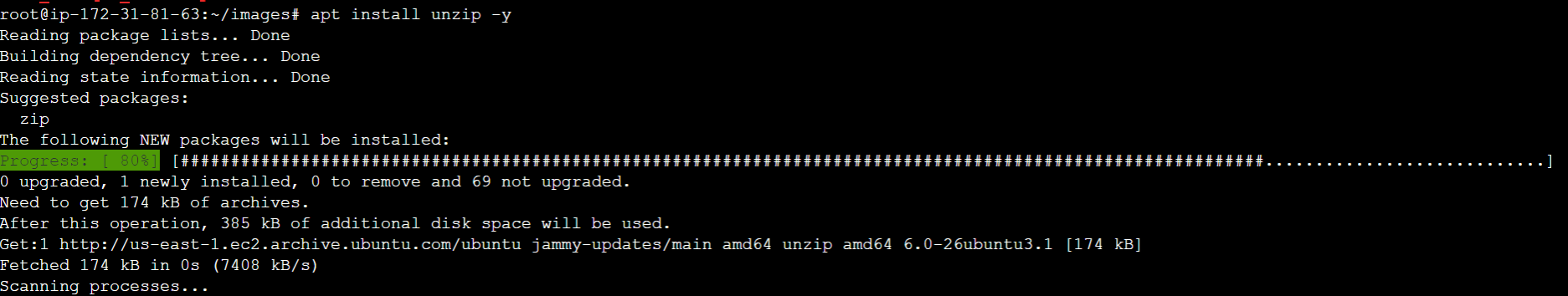
1. Get a sample webApp, used tooplate.com in this example & get the downloadable url using inspect element :

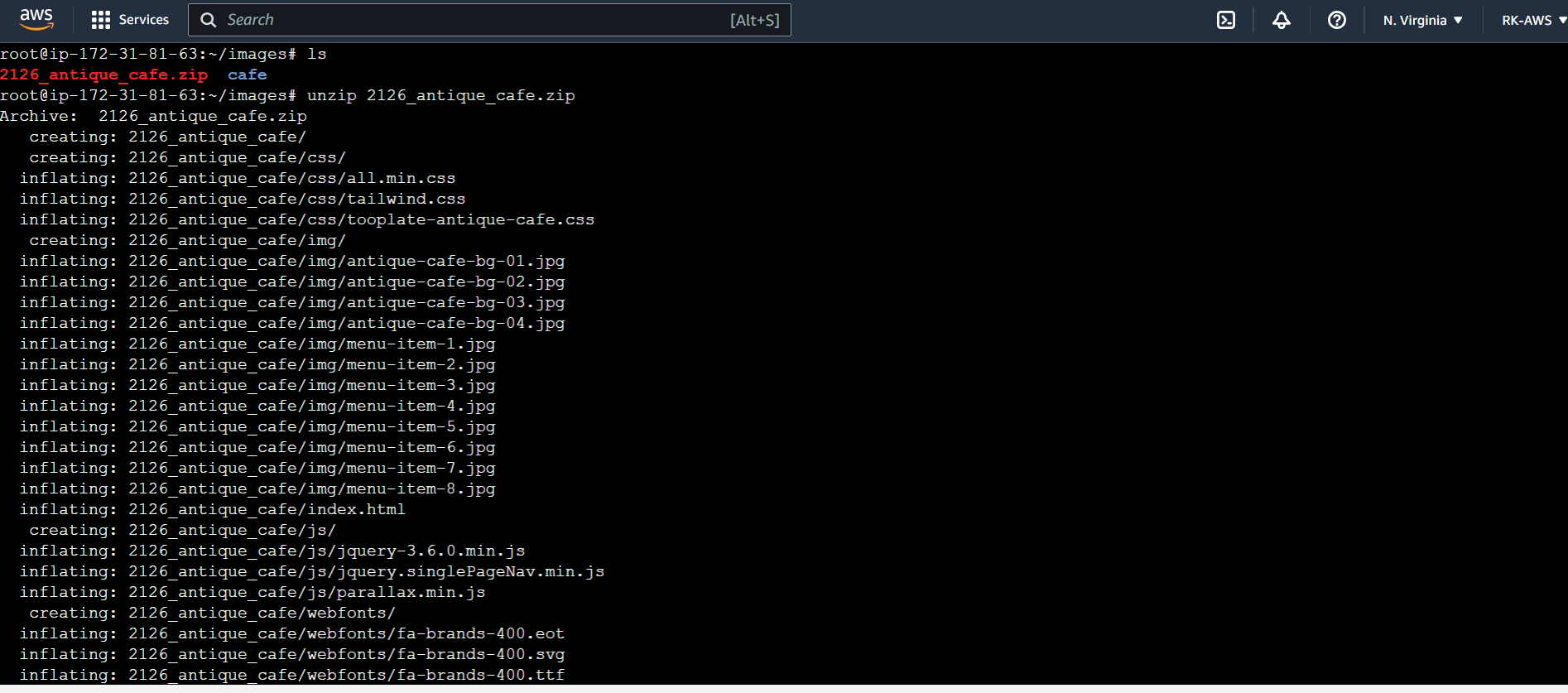


<https://www.tooplate.com/zip-templates/2126_antique_cafe.zip>

1. Download the above zip file in the project directory created above & unzip it. Then archive it.

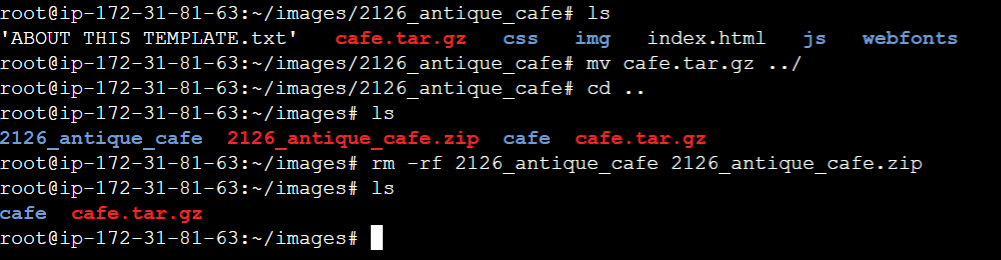




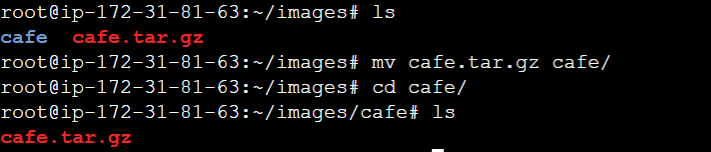




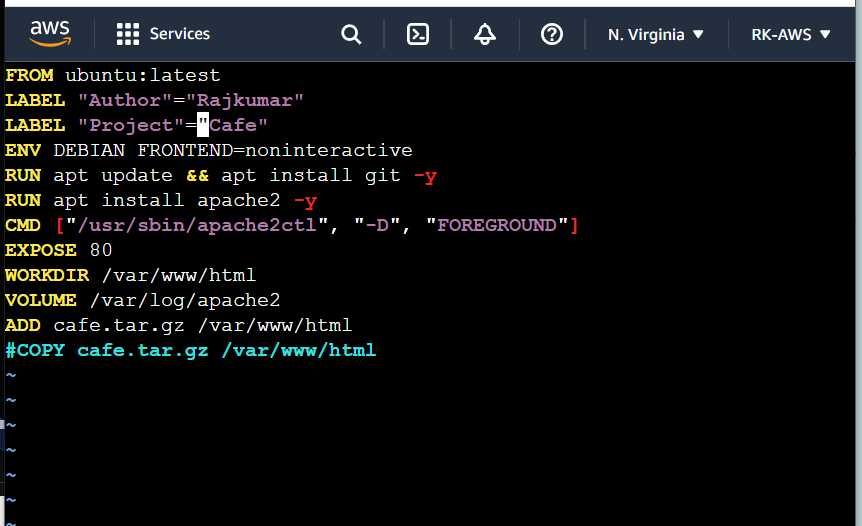
Move tar file 1 level up & remove other zip file & unzipped folder :



Move the tar file (artifact) to the café project directory :



1. Write the docker file :



FROM – will contain information of base image

LABEL – is just like giving tags in AWS

ENV – this is to provide environment variable, in this case to avoid interactive run

RUN – instructions to perform some tasks like installation or update package, this can be doneon multiple lines or can be done on same line using ‘&&’. Its advisible to have multiple commands in single line as every RUN there will be a layer created.

CMD – this is to start the process, in this case as we cannot start the service, we need rely on binary, so we need to pass the location of the binary, along with some arguments required

EXPOSE – this is to tell which port will be used by the service.

WORKDIR – this will be directory where a user will be able to run commands after getting attached to the container.

VOLUME – required if want to preserve anything if container is deleted, in this case it will store the logs

ADD – untar the contents & copy

COPY – will simply copy



