

## Module 8: React Application Testing and Deployment

---

### Demo Document 3

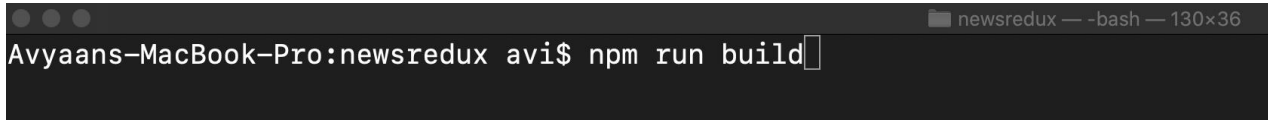
edureka!

**edureka!**

© Brain4ce Education Solutions Pvt. Ltd.

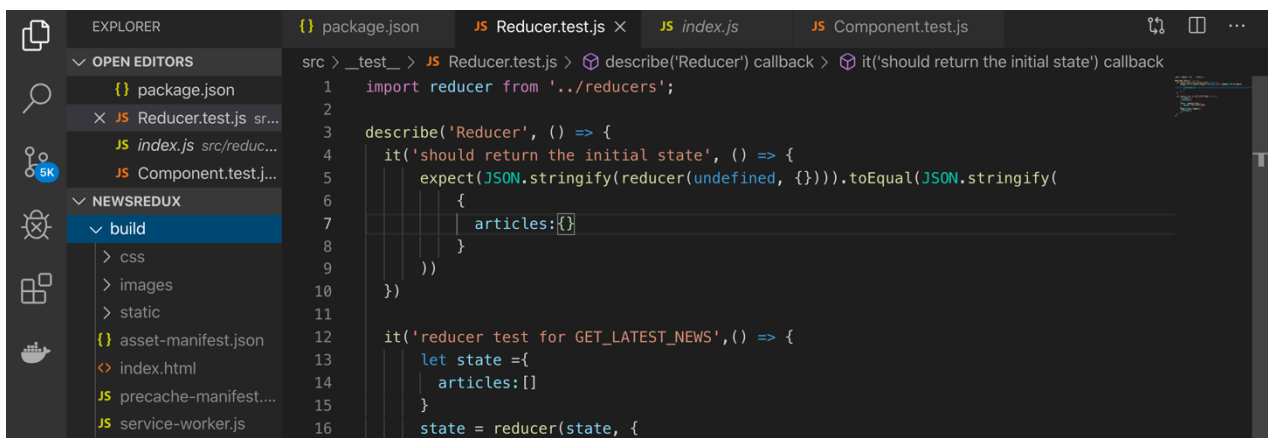
## Running your React application on production server: Nginx

**Step 1:** To take application in production we need to create the build of the app using npm run build.



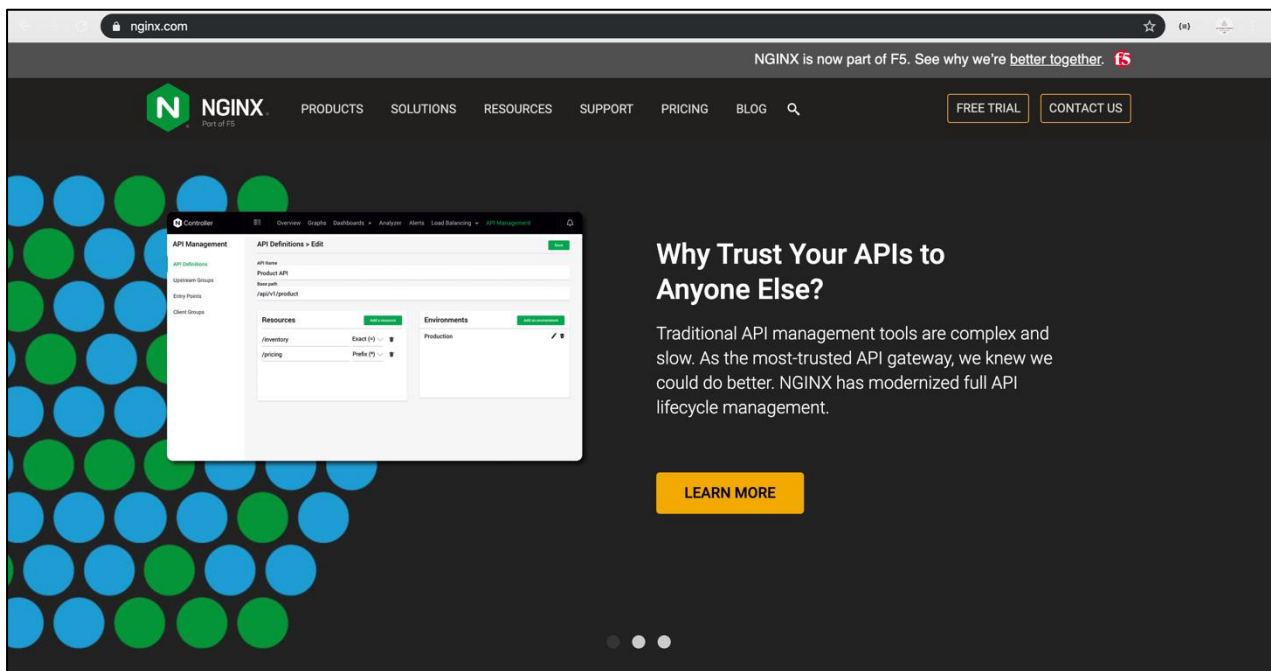
```
newsredux — -bash — 130x36
Aavyaans-MacBook-Pro:newsredux avi$ npm run build
```

**Step 2:** This will create the build folder in the main application folder. We have to store build in root location from where we can boot the application.



**Step 3:** Create a root folder www, inside it create another folder called react-app and place your build file in it. (mac users can create root folder in var section and windows user can create this root folder in c drive)

**Step 4:** Install the Nginx as production server in our system. (make use of this link to download the Nginx: <http://nginx.org/en/download.html>)



**Step 5:** In the Nginx directory open conf file, choose nginx file add server configuration like on which port we have to run the application and paste the directory for the root folder to pick the application .

At last we have to type 'nginx' to make the application run over the Nginx server.

```
/
#error_page 404          /404.html;

# redirect server error pages to the static page /50x.html
#
error_page 500 502 503 504 /50x.html;
location = /50x.html {
    root   html;
}
#location ~ /\.php$ {
#    proxy_pass http://127.0.0.1;
#}

# proxy the PHP scripts to Apache listening on 127.0.0.1:80
#

# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
#
#location ~ /\.php$ {
#    root           html;
#    fastcgi_pass   127.0.0.1:9000;
#    fastcgi_index  index.php;
#    fastcgi_param  SCRIPT_FILENAME /scripts$fastcgi_script_name;
#    include        fastcgi_params;
#}

# deny access to .htaccess files, if Apache's document root
# concurs with nginx's one
#
#location ~ /\.ht {
#    deny all;
#}
}

# another virtual host using mix of IP-, name-, and port-based configuration
#
server {
    listen      8081;
    server_name localhost;

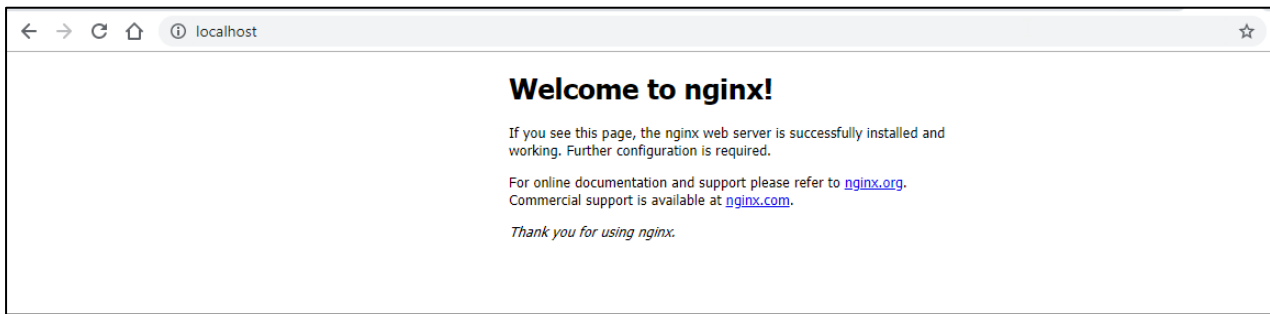
    location / {
        root   /var/www/react-app;
        index  index.html index.htm;
    }
}

server {
    listen      8082;
    server_name localhost;

    location / {
        proxy_pass http://127.0.0.1:8900;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}

server {
    listen 8089;
    server_name localhost;
```

**Step 6:** Check your localhost verify Nginx is working.



**Step 7:** Check the localhost 8082 to see working of application on Nginx server.



**Conclusion:**

We have successfully performed the deployed the React application on Nginx server.

edureka!