Leopard\_solutions setup

**SSH Key Authentication(One Time Setup):**

When we need to access any remote server and to execute any commands on it we need to authenticate the ssh key, so that we will have access to execute any commands on remote server or client server.

Login to jenkins server and login as jenkins user using the command

**sudo su jenkins -s /bin/bash** and change the path to generate ssh-key using the cmd

**cd /var/lib/jenkins/.ssh/.**

After reaching above path , please generate the ssh-key using ssh-keygen cmd

Here it asks for the key name , please enter name and password here .

After generation of ssh keys will see both public and private keys in the path /home/guest/.ssh (**cd /var/lib/jenkins/.ssh/**).

The public key is in name.pub format and the private key is just the name which you have entered while generating the key.

.

Now copy the public key key to client server using below command

**ssh-copy-id pubkey remote\_user@host\_ip** .

Now verify whether we are able to login client server using the cmd

**ssh -i private\_key remote\_user@host\_ip**.

Now update the private key which was genterate to earlier into the jenkins console. In manage jenkins and select credentials then > select ssh userName with private key from drop down.

Now select the private key from credentials and add our private key there and click on create.

jenkins command line execution command for syncing code from repository

**Jenkins suite for leopard**

1. To publish the code we use the below command

**dotnet publish /var/lib/jenkins/jobs/leopard\_solutions\_deployment/workspace/LeoPard\_Linux\_Project -f net6.0 -c Release**

1. By using the above, we sync updated code to jenkins and server and publishing the code with below command and after publishing the update code , we are syncing the .dll files to the host\_server path.

**rsync -avzh --exclude='\*.svn' --no-perms -og --chown=root:root /var/lib/jenkins/jobs/leopard\_solutions\_deployment/workspace/LeoPard\_Linux\_Project/bin/Release/net6.0/publish/ root@10.160.0.3:/var/www/build/leopard\_solutions/**

By executing the above command we will convert .dll to release where the code is executed

3) Now we will run the marketplace as service by creating new file using below :

**nano /etc/systemd/system/leopard\_solutions.service**

**[Unit]**

**Description=leopard\_solutions\_api**

**[Service]**

**WorkingDirectory=/var/www/build/leopard\_solutions/**

**ExecStart=/usr/bin/dotnet /var/www/build/leopard\_solutions/LeoPard\_Linux\_Project.dll --urls=**[**http://0.0.0.0:5005**](http://0.0.0.0:5005/)

**Restart=always**

**RestartSec=10**

**SyslogIdentifier=test**

**User=root**

**Environment=ASPNETCORE\_ENVIRONMENT=Production**

**Environment=DOTNET\_PRINT\_TELEMETRY\_MESSAGE=true**

**[Install]**

**WantedBy=multi-user.target**

4) Commands to start the service which we have done

**service leopard\_solutions start**

**service leopard\_solutions status**

5) Create the config file for market service in nginx file configurations

path to create conf file

**nano /etc/nginx/site-available/marketplace.conf**

**server {**

**listen 443 ssl;**

**listen [::]:443 ssl;**

**server\_name attorney.ceipal.com;**

**ssl on;**

**ssl\_certificate /etc/ssl/certs/2020-ceipal.com.bundle.crt;**

**ssl\_certificate\_key /etc/ssl/certs/2020-ceipal.com.key;**

**ssl\_session\_cache shared:SSL:5m;**

**ssl\_session\_timeout 10m;**

**ssl\_protocols TLSv1.2;**

**ssl\_ciphers 'ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!3DES:!MD5:!PSK:!SHA1:!DSS';**

**ssl\_prefer\_server\_ciphers on;**

**access\_log /var/log/nginx/marketplacestg.ceipal.com-ssl-access.log;**

**error\_log /var/log/nginx/marketplacestg.ceipal.com-ssl-error.log;**

**include /etc/nginx/mime.types;**

**root /var/www/build/leopard\_solutions/wwwroot;**

**location /documents {**

**alias /var/www/build/leopard\_solutions/wwwroot;**

**}**

**# Add index.php to the list if you are using PHP**

**index index.html index.htm index.nginx-debian.html;**

**server\_name marketplacestg.ceipal.com;**

**location / {**

**# First attempt to serve request as file, then**

**# as directory, then fall back to displaying a 404.**

**try\_files $uri $uri/ =404;**

**proxy\_pass http://127.0.0.1:5005;**

**# proxy\_pass\_reverse / http://127.0.0.1:5001/;**

**include /etc/nginx/sites-enabled/marketplace.conf;**

**proxy\_http\_version 1.1;**

**proxy\_set\_header Upgrade $http\_upgrade;**

**proxy\_set\_header Connection keep-alive;**

**proxy\_set\_header Host $host;**

**proxy\_cache\_bypass $http\_upgrade;**

**proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;**

**proxy\_set\_header X-Forwarded-Proto $scheme;**

**}**

**# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000**

**#**

**#location ~ \.php$ {**

**# include snippets/fastcgi-php.conf;**

**#**

**# # With php7.0-cgi alone:**

**# fastcgi\_pass 127.0.0.1:9000;**

**# # With php7.0-fpm:**

**# fastcgi\_pass unix:/run/php/php7.0-fpm.sock;**

**#}**

**# deny access to .htaccess files, if Apache's document root**

**# concurs with nginx's one**

**#**

**#location ~ /\.ht {**

**# deny all;**

**#}**

**}**

**To enable the site from sites-available to sites-enabled**

**ln -s /etc/nginx/sites-available/leopard\_solutions.conf /etc/nginx/sites-enabled/**

6) restart the nginx service and check the rendering pages.

**service nginx restart**

**server nginx status**

**Note : -** After doing every deployment please execute the command which is mentioned below

**service leopard\_solutions restart**