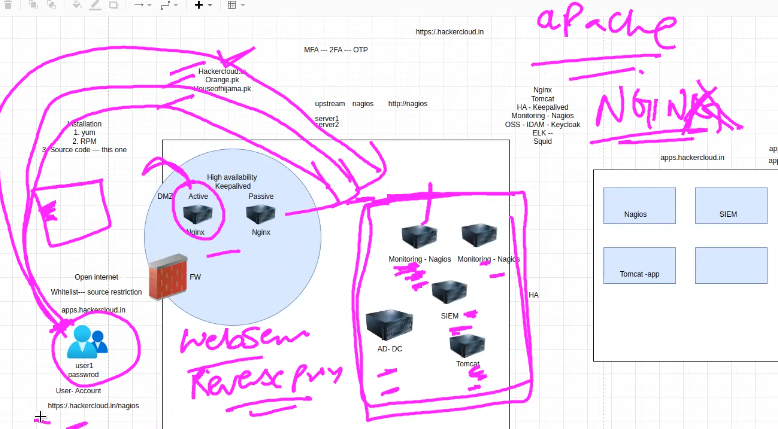
**Lecture 28**

Installing Reverse Proxy Nginx Cluster (OSS 1.20.2) to protect our Internal assets Part2 (Live Session 13 November 2022)

Nginx 🡪

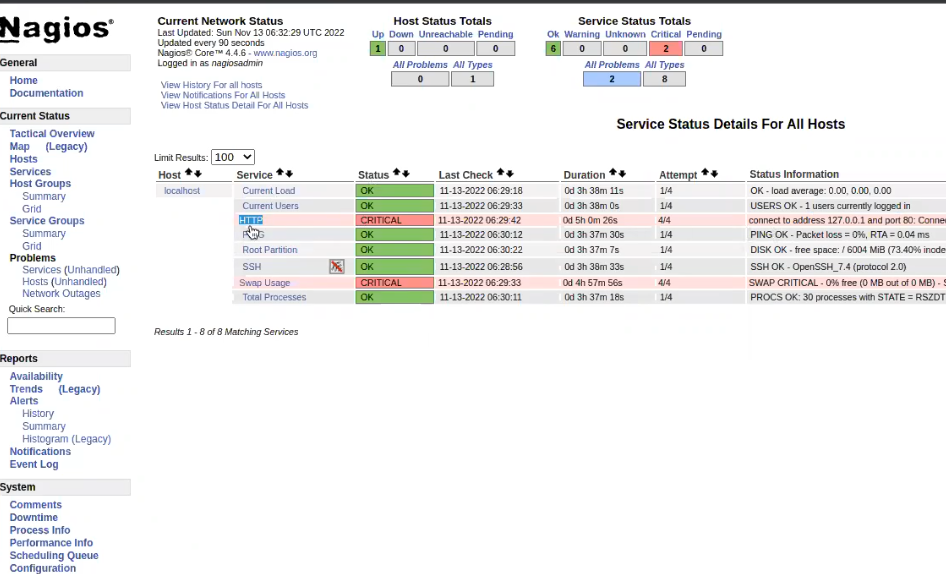
* Web Server
* Reverse proxy (Nginx's reverse proxy functionality is achieved using its **proxy\_pass** directive. When a request is received by the Nginx server, the **proxy\_pass** directive forwards the request to a specified backend server, which could be another web server or application server.)



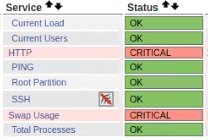
-

<http://hackercloud.in> 🡪 landing page. (main URL)

<http://hackercloud.in/nagios> 🡪 Nginx will retrieve a web page (specific) and after authentication.

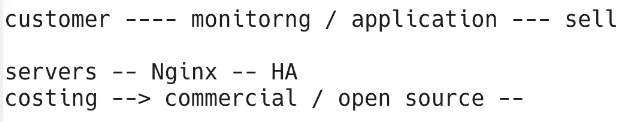


Interview prep 🡪 we have a monitoring tool, we monitor all the servers with Nagios 🡪 CPU, Memory, Disk usage 🡪



-

Project for the company



Recommendation 🡪 commercial version of Nginx or Open Source.



How to decide?

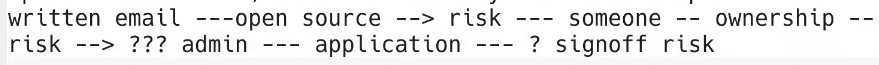
How to fund the project? 🡪 if the company can earn then decide to spend.

SLA (Service Level Agreement) 99.999% 🡪 depends upon available bandwidth.

 🡪 risk impact in case of vulnerability 🡪 we have to wait until the patch is ruled out for that particular vulnerability.

**zero-day** refers to a software vulnerability or security flaw that is unknown to the software vendor or developers, and for which no patch or fix has been released. Attackers can exploit zero-day vulnerabilities to launch cyber attacks, as there is no known defense or fix available to protect against the attack.

WARNING!!!



It should be on written (send an email) even if CEO decides to go for open source.

Don’t own the risk!!! As we are not owner os the server 🡪 we are “data custodian””

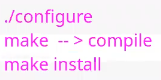
-

Open source 🡪 installation

We have decided to install it through **“source Code.”**

HA in case 🡪 SLA 99.999% availability.

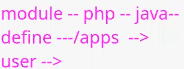
It depends upon requirement.



Benefits of Source Code installation

Lightweight

Customized 🡪 modules 🡪



.

Drawback of Source Code installation.

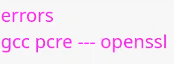
Maintenance is difficult.

Systemctl start nginx 🡪we have to create unit file manually. (Unit file in system)

|  |
| --- |
| **how to create unit file in system for nginx ?**  To create a unit file for Nginx on a systemd-based Linux system, you can follow the steps below:   1. Open a text editor on your Linux machine. You can use any text editor of your choice, such as nano or vim. 2. Create a new file with the name "nginx.service" in the directory "/etc/systemd/system/". For example, you can use the following command to create the file using nano editor:   sudo nano /etc/systemd/system/nginx.service   1. Paste the following content into the file:   [Unit]  Description=The NGINX HTTP and reverse proxy server After=syslog.target network.target remote-fs.target nss-lookup.target  [Service]  Type=forking PIDFile=/run/nginx.pid ExecStartPre=/usr/sbin/nginx -t -q -g 'daemon on; master\_process on;' ExecStart=/usr/sbin/nginx -g 'daemon on; master\_process on;' ExecReload=/bin/kill -s HUP $MAINPID ExecStop=/bin/kill -s TERM $MAINPID  [Install]  WantedBy=multi-user.target   1. Save and close the file. 2. Reload the systemd configuration to make it aware of the new unit file:   sudo systemctl daemon-reload   1. Enable the Nginx service to start at boot:   sudo systemctl enable nginx.service   1. Start the Nginx service:   sudo systemctl start nginx.service  That's it! Your Nginx server should now be running and automatically starting up on boot. You can also manage the service using systemctl commands, such as **systemctl stop nginx.service** to stop the service, **systemctl restart nginx.service** to restart it, and so on. |

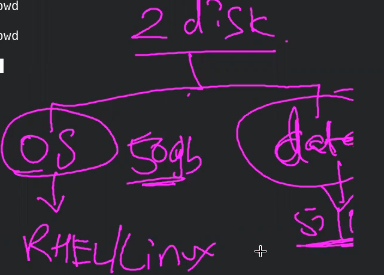
Errors may occur during source code installation.

Dependencies



-

Installation



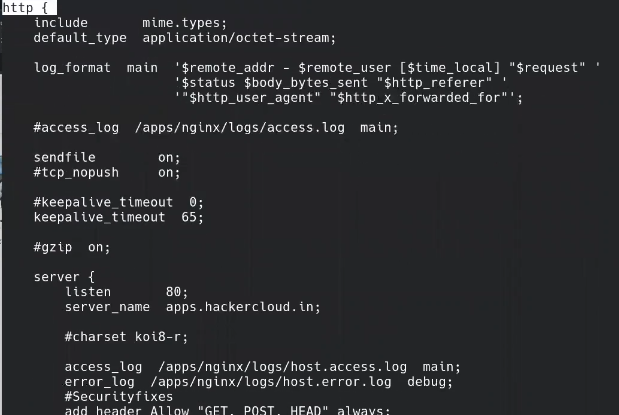
LVM would be a best option in case the date is increasing on that drive.

-

Config file 🡪

3 sessions

http 🡪 global settings



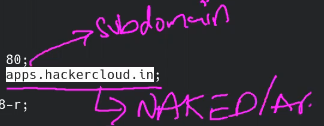
-

server 🡪 define server’s name

Text

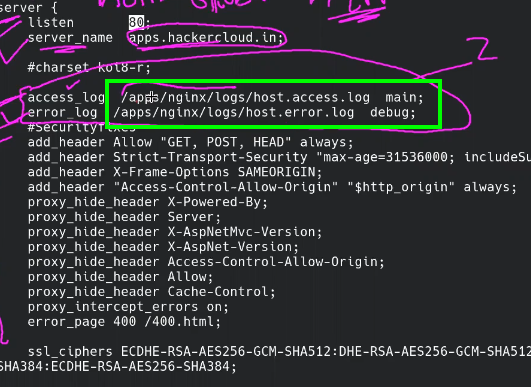
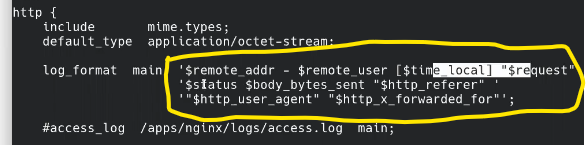
Description automatically generated

.



.

In server block we can define where to save our “logs”

* If we define it in server block 🡪 it will only show logs related to that specific server
* 
* But if we create logs in “http” (global block) 🡪 we can have logs related to whole web site (or whole domain) 🡪 log file should be created with the specific server name I.e clients.hacekercloud.in, apps.hackercloud.in instead of error.log or access.log
* .
* Log format can be improved in “http” global section in conf file ,
* 
* .
* Logs are being generated.

***What is the best option???***

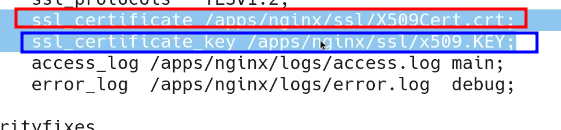
* ***Logically*** *🡪 single log file for all servers will keep increasing inside the directory.*
* *It will be difficult to investigate in case of a single file 🡪 it will mix up everything.*

In config file there is SSL protocol field which can be set accordingly 🡪 TLSv1.1 is not recommended.

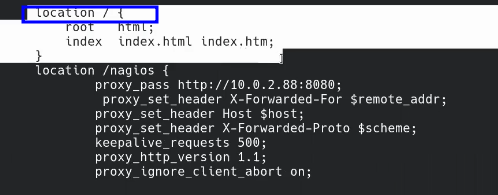


.

SL certificate and its key also mentioned in there,

.

🡪 location section under server section

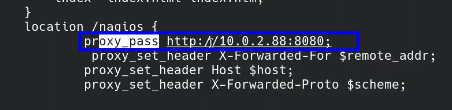


Index.html page’s location

-

3rd section 🡪 location section / location block

Here we can define the “reverse proxy”



Proxy\_pass

The url <http://10.0.2.88:8080> is Nagios server address.

This is private IP which is in private network.

As shown in the fig we were discussion.

***Every server in the private network will have this location block with its local IP address.***

.

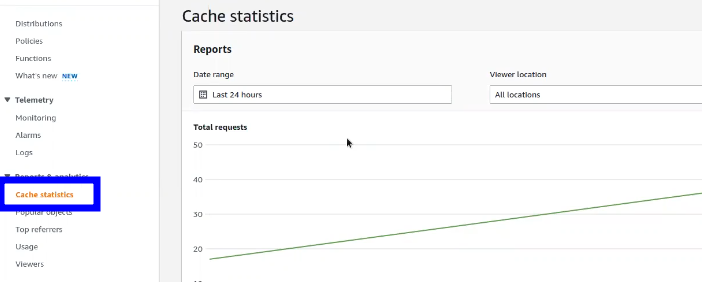
Tip:- Sir Kazim is using S3 bucket for his website where index.html is saved, but AWS charges for a certain hits 🡪 to save hits Sir Kazim have “cached” his webpage on “cloudflare” “edge location” which saved hits to S3 bucket

. Diagram

Description automatically generated

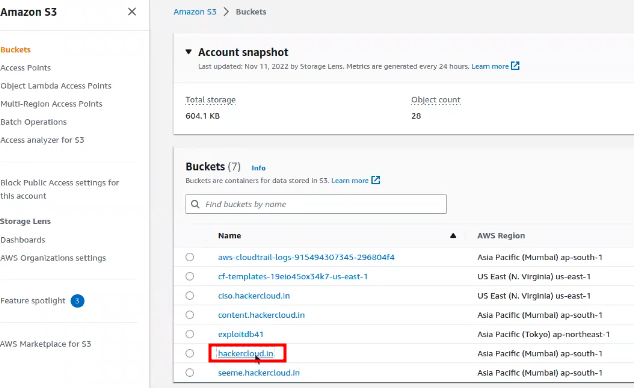
.

***AWS has Cache status service***



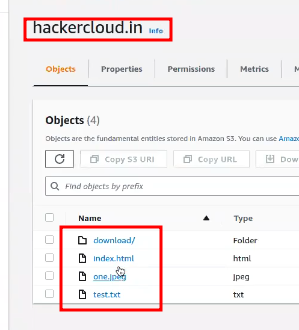
.

To host a web page on S3 🡪 create S3 bucket as the same name of your domain i.e hackercloud.in



.

Then all files are saved in that S3 bucket.



.

Then enable ***Static web hosting*** in S3

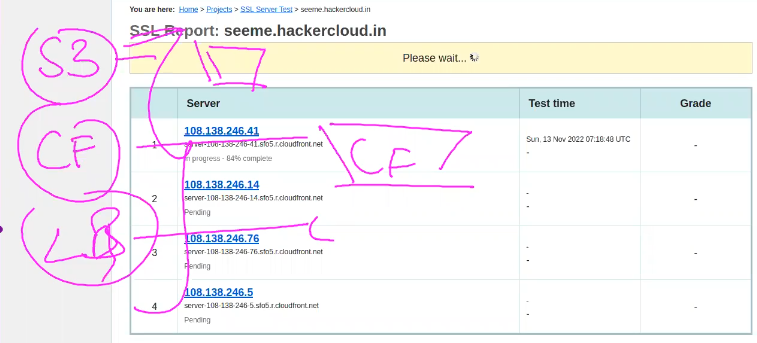
* It will provide a website link
* If your domain is registered 🡪 AWS provides rout 53 service that works as a DNS.
* Here Sir Kazim have configure cloudfrom edge location so that his page is cached on “edge location” and save hits to S3 bucket.

.

We need to provide the access to our clients online through “https” 🡪 SSL certificate is our requirement 🡪 how to get SSL ?

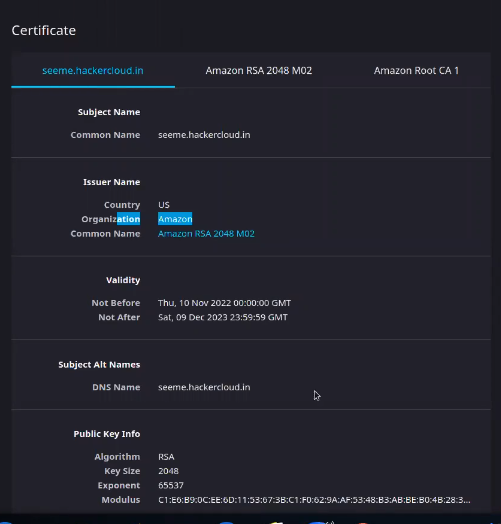
1 way to generate self-signed SSL certificate but it will show warning in the browser.

If the service is in AWS 🡪 AWS Certificate Manager 🡪 free 🡪 for S3 bucket 🡪 not for EC2 instance.

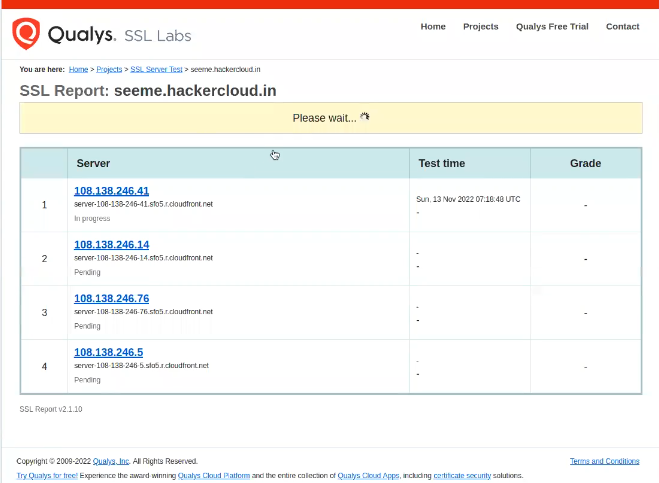


.

Sir Kazim have used it 🡪 seeme.hackercloud.in



.



.

Another great option – **certbot**

Provides free SSL

**Certbot** is a free, open-source software tool for automatically configuring and renewing SSL/TLS certificates on web servers. It is part of the **Let's Encrypt project**, which provides free SSL/TLS certificates that are trusted by most modern web browsers.

Letsencrypt.org 🡪 millions of SSL certificates 🡪 industry use it to save cost.

It is “**command line”** based

-

Using port 80 is not recommended.

Port 443 is to be used

* Sefl signed SSL 🡪 not recommended for external client
* Wild card certificate \*.<domain> \*.hackercloud.in
  + Or <subdomain>.<domain>. 🡪 client.hackercloud.in
* **certbot** 🡪 domain must be registered publicly.