Smart HTTP सेटअप

Smart HTTP Git क्लाइन्टलाई प्रमाणीकृत र अप्रमाणीकृत पहुँच प्रदान गर्न प्रयोग गरिन्छ। यसलाई **git-http-backend** नामक CGI स्क्रिप्टको मद्दतले कन्फिगर गर्न सकिन्छ। यो सेटअप विशेष गरी **Apache** प्रयोग गरेर देखाइएको छ।

**१. Apache स्थापना गर्नुहोस्**

यदि तपाईंसँग Apache सेटअप छैन भने, निम्न कमाण्ड प्रयोग गरेर स्थापना गर्नुहोस्:

$ sudo apt-get install apache2 apache2-utils

**२. आवश्यक मोड्युल सक्षम गर्नुहोस्**

Apache मा CGI, Alias, र Env मोड्युलहरू सक्षम गर्न:

$ sudo a2enmod cgi alias env

**३. Git HTTP-Backend कन्फिगर गर्नुहोस्**

Apache ले Git HTTP-Backend सम्हाल्न सक्छ। निम्न चरणहरू पालना गर्नुहोस्:

1. Apache को CGI-सक्रिय डाइरेक्टरीमा Git HTTP-Backend का लागि स्क्रिप्टको पथ सेटअप गर्नुहोस्।
2. **Apache कन्फिगरेसन फाइल सम्पादन गर्नुहोस्।**  
   /etc/apache2/sites-available/000-default.conf फाइल खोल्नुहोस्:

$ sudo nano /etc/apache2/sites-available/000-default.conf

1. यो सामग्री थप्नुहोस्:

SetEnv GIT\_PROJECT\_ROOT /var/www/git

SetEnv GIT\_HTTP\_EXPORT\_ALL

ScriptAlias /git/ /usr/lib/git-core/git-http-backend/

<Directory "/usr/lib/git-core">

Options ExecCGI

Require all granted

</Directory>

* + **GIT\_PROJECT\_ROOT**: Git रिपोजिटरीहरूको आधार पथ।
  + **GIT\_HTTP\_EXPORT\_ALL**: सबै प्रोजेक्टहरूलाई एक्सपोर्ट गर्न अनुमति दिन्छ।
  + **ScriptAlias**: /git/ URL लाई CGI स्क्रिप्टमा पुनर्निर्देशित गर्दछ।

**४. रिपोजिटरीको स्थान तयार गर्नुहोस्**

Git रिपोजिटरीहरू /var/www/git मा राख्न:

$ sudo mkdir -p /var/www/git/project.git

$ sudo git init --bare /var/www/git/project.git

$ sudo chown -R www-data:www-data /var/www/git

**५. Apache पुनःसुरु गर्नुहोस्**

कन्फिगरेसन लागू गर्न Apache पुनःसुरु गर्नुहोस्:

$ sudo systemctl restart apache2

**६. Smart HTTP को प्रयोग**

अब तपाईँले निम्न प्रकारले रिपोजिटरी क्लोन गर्न सक्नुहुन्छ:

$ git clone http://<your-server-ip>/git/project.git

**७. प्रमाणीकरण कन्फिगर गर्नुहोस् (वैकल्पिक)**

यदि प्रमाणीकरण आवश्यक छ भने, Apache मा **Basic Authentication** सक्षम गर्न सकिन्छ।

1. एक पासवर्ड फाइल सिर्जना गर्नुहोस्:

$ sudo htpasswd -c /etc/apache2/.htpasswd username

1. Apache कन्फिगरेसन फाइलमा प्रमाणीकरण नियम थप्नुहोस्:

<Location "/git">

AuthType Basic

AuthName "Restricted Access"

AuthUserFile /etc/apache2/.htpasswd

Require valid-user

</Location>

1. Apache पुनःसुरु गर्नुहोस्:

$ sudo systemctl restart apache2

अब Smart HTTP ले प्रमाणीकरण सहित काम गर्नेछ।

Smart HTTP ले Git मा **authenticated** र **unauthenticated** दुबै प्रकारको पहुँच प्रदान गर्दछ। यसमा Git ले प्रदान गर्ने **git-http-backend** CGI स्क्रिप्टको प्रयोग गरिन्छ। तल Apache सर्भर प्रयोग गरेर Smart HTTP सेटअप गर्ने विस्तृत प्रक्रिया छ।

**१. आवश्यक तयारी**

सबैभन्दा पहिले, Apache सर्भर र यसको लागि आवश्यक मोड्युलहरू स्थापना गर्नुहोस्:

$ sudo apt-get install apache2 apache2-utils

$ sudo a2enmod cgi alias env

यो कमाण्डले Apache मा CGI (**mod\_cgi**), Alias (**mod\_alias**) र Env (**mod\_env**) मोड्युलहरू सक्रिय गर्छ।

**२. Git डाइरेक्टरीको अनुमति परिवर्तन**

Apache सर्भरलाई Git डाइरेक्टरीमा पढ्ने/लेख्ने अधिकार दिन आवश्यक छ। यसका लागि /srv/git डाइरेक्टरीको Unix समूहलाई www-data मा परिवर्तन गर्नुहोस्:

$ sudo chgrp -R www-data /srv/git

**३. Apache कन्फिगरेसन सम्पादन**

Apache कन्फिगरेसन फाइल (जस्तै /etc/apache2/sites-available/000-default.conf) सम्पादन गरी तलको ब्लक थप्नुहोस्:

SetEnv GIT\_PROJECT\_ROOT /srv/git

SetEnv GIT\_HTTP\_EXPORT\_ALL

ScriptAlias /git/ /usr/lib/git-core/git-http-backend/

<Directory "/usr/lib/git-core">

Options ExecCGI

Require all granted

</Directory>

<Files "git-http-backend">

AuthType Basic

AuthName "Git Access"

AuthUserFile /srv/git/.htpasswd

Require expr !(%{QUERY\_STRING} -strmatch '\*service=git-receive-pack\*' || %{REQUEST\_URI} =~ m#/git-receive-pack$#)

Require valid-user

</Files>

**कन्फिगरेसनको विवरण:**

* **SetEnv GIT\_PROJECT\_ROOT**: Git को सबै रिपोजिटोरीहरूको रुट डाइरेक्टरी निर्धारण गर्दछ।
* **SetEnv GIT\_HTTP\_EXPORT\_ALL**: सबै रिपोजिटोरीहरूलाई बिना git-daemon-export-ok फाइलको पनि Git सर्भरले सेवा दिन सक्षम बनाउँछ।
* **ScriptAlias**: /git URL बाट आउने अनुरोधलाई git-http-backend CGI स्क्रिप्टतर्फ पठाउँछ।
* <Files> ब्लक:
  + **AuthType Basic**: आधारभूत HTTP प्रमाणीकरण सक्रिय गर्छ।
  + **AuthUserFile**: प्रमाणीकरणका लागि प्रयोगकर्ताहरूको पासवर्ड समावेश गरिएको .htpasswd फाइल निर्धारण गर्छ।
  + **Require valid-user**: Git मा **write** अपरेशनका लागि प्रमाणीकरण आवश्यक पर्छ।

**४. प्रमाणीकरणको लागि .htpasswd फाइल सिर्जना गर्नुहोस्**

प्रयोगकर्ताहरूको प्रमाणीकृत पहुँचको लागि .htpasswd फाइल बनाउनुहोस्:

$ sudo htpasswd -c /srv/git/.htpasswd username

**थप प्रयोगकर्ता थप्न (बिना फाइल मेट्ने):**

$ sudo htpasswd /srv/git/.htpasswd another\_user

**५. SSL सक्रिय गर्नुहोस् (डेटा इन्क्रिप्ट गर्न)**

HTTP को सट्टा HTTPS प्रयोग गरी डेटा सुरक्षित बनाउन SSL सेटअप गर्नुहोस्।

**SSL मोड्युल स्थापना र Apache पुनःसुरु गर्नुहोस्:**

$ sudo a2enmod ssl

$ sudo systemctl restart apache2

**SSL कन्फिगरेसन थप्नुहोस्**

SSL कन्फिगरेसनका लागि /etc/apache2/sites-available/default-ssl.conf सम्पादन गर्नुहोस्:

<VirtualHost \*:443>

ServerAdmin admin@example.com

DocumentRoot /var/www/html

SSLEngine on

SSLCertificateFile /path/to/your/certificate.crt

SSLCertificateKeyFile /path/to/your/private.key

<Location /git>

SetEnv GIT\_PROJECT\_ROOT /srv/git

SetEnv GIT\_HTTP\_EXPORT\_ALL

ScriptAlias /git/ /usr/lib/git-core/git-http-backend/

<Files "git-http-backend">

AuthType Basic

AuthName "Git Access"

AuthUserFile /srv/git/.htpasswd

Require expr !(%{QUERY\_STRING} -strmatch '\*service=git-receive-pack\*' || %{REQUEST\_URI} =~ m#/git-receive-pack$#)

Require valid-user

</Files>

</Location>

</VirtualHost>

**SSL सक्रिय गर्ने र साइट सक्षम गर्ने:**

$ sudo a2ensite default-ssl

$ sudo systemctl reload apache2

**६. प्रमाणीकरणको परीक्षण**

* **Git Clone (Unauthenticated Read):**

$ git clone http://your-server-ip/git/repository-name.git

* **Git Push (Authenticated Write):**

$ git push http://your-server-ip/git/repository-name.git

प्रमाणीकरणका लागि .htpasswd मा रहेको प्रयोगकर्ता र पासवर्ड आवश्यक पर्छ।

**यो सेटअपले HTTP मार्फत Smart Git सेवा सुरु गर्छ जसले fetch र push दुवैलाई समर्थन गर्दछ। SSL को प्रयोगले सेवा अझै सुरक्षित बनाउँछ।**