सर्भर सेटअप गर्ने तरिका (नेपालीमा)

SSH एक्सेस सर्भर साइडमा सेटअप गर्ने प्रक्रिया यसरी हुन्छ। यस उदाहरणमा, प्रयोगकर्ताहरूलाई प्रमाणित गर्नको लागि **authorized\_keys** विधि प्रयोग गरिएको छ। यहाँ मानिएको छ कि तपाईँले सामान्य लीनक्स वितरण (जस्तै, Ubuntu) प्रयोग गर्दै हुनुहुन्छ।

**महत्त्वपूर्ण सूचना:**

केही प्रक्रियाहरूलाई **ssh-copy-id** कमाण्ड प्रयोग गरेर स्वचालित गर्न सकिन्छ। यसले म्यानुअली सार्वजनिक की कपी र इन्स्टल गर्नु पर्ने झन्झटलाई हटाउँछ।

**१. git प्रयोगकर्ता खाता सिर्जना गर्नुहोस् र .ssh डाइरेक्टरी सेटअप गर्नुहोस्।**

$ sudo adduser git

$ su git

$ cd

$ mkdir .ssh && chmod 700 .ssh

$ touch .ssh/authorized\_keys && chmod 600 .ssh/authorized\_keys

* **adduser git**: नयाँ git प्रयोगकर्ता खाता सिर्जना गर्छ।
* **.ssh डाइरेक्टरी बनाउने**: यो डाइरेक्टरीमा SSH कीहरू सुरक्षित तरिकाले राख्न सकिन्छ।
* **फाइल अनुमति सेट गर्ने**:
  + **chmod 700 .ssh**: .ssh डाइरेक्टरीलाई मात्र मालिकले पहुँच गर्न मिल्ने बनाउँछ।
  + **chmod 600 authorized\_keys**: authorized\_keys फाइललाई सुरक्षित बनाउँछ।

**२. सार्वजनिक कीहरू authorized\_keys फाइलमा थप्नुहोस्।**

मानौं तपाईंसँग केही विकासकर्ताहरूका विश्वसनीय सार्वजनिक कीहरू छन्, र ती अस्थायी फाइलहरूमा सुरक्षित छन्। ती कीहरू authorized\_keys फाइलमा थप्न सकिन्छ।

**उदाहरण:**

$ cat /tmp/id\_rsa.john.pub

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCB007n/ww+ouN4gSLKssMxXnBOvf9LGt4L

ojG6rs6hPB09j9R/T17/x4lhJA0F3FR1rP6kYBRsWj2aThGw6HXLm9/5zytK6Ztg3RPKK+4k

Yjh6541NYsnEAZuXz0jTTyAUfrtU3Z5E003C4oxOj6H0rfIF1kKI9MAQLMdpGW1GYEIgS9Ez

Sdfd8AcCIicTDWbqLAcU4UpkaX8KyGlLwsNuuGztobF8m72ALC/nLF6JLtPofwFBlgc+myiv

O7TCUSBdLQlgMVOFq1I2uPWQOkOWQAHukEOmfjy2jctxSDBQ220ymjaNsHT4kgtZg2AYYgPq

dAv8JggJICUvax2T9va5 gsg-keypair

**यसलाई authorized\_keys मा थप्नुहोस्:**

$ cat /tmp/id\_rsa.john.pub >> ~/.ssh/authorized\_keys

यो कमाण्डले सार्वजनिक कीलाई authorized\_keys फाइलमा जोड्छ।

**३. प्रक्रिया पूरा भएपछि git प्रयोगकर्ताले SSH मार्फत पहुँच गर्न सक्छ।**

अब, सबै विकासकर्ताले आफ्नो सार्वजनिक की पठाउनु पर्छ। त्यसपछि उनीहरूको की authorized\_keys फाइलमा थप्नुपर्छ। यो प्रक्रियाले सर्भरमा सुरक्षित र कुशल पहुँच सुनिश्चित गर्दछ।

**टिप्स:**

* **की को सुरक्षा:**  
  सुनिश्चित गर्नुहोस् कि केवल विश्वसनीय सार्वजनिक कीहरू मात्र authorized\_keys मा राखिएको छ।
* **ssh-copy-id को प्रयोग:**  
  विकासकर्ताहरूले यो उपकरण प्रयोग गरेर स्वचालित रूपमा आफ्नो सार्वजनिक की सर्भरमा थप्न सक्छन्।

अब, खाली रिपोजिटरी सेटअप गर्न git init लाई --bare विकल्पसँग चलाउनुहोस्, जसले बिना वर्किङ डाइरेक्टरी रिपोजिटरी इनिसियलाइज गर्दछ:

$ cd /srv/git

$ mkdir project.git

$ cd project.git

$ git init --bare

Initialized empty Git repository in /srv/git/project.git/

अब, John, Josie, वा Jessica ले आफ्नो प्रोजेक्टको पहिलो संस्करणलाई उक्त रिपोजिटरीमा पुश गर्न सक्छन्, त्यसलाई रिमोटको रूपमा थपेर र ब्रान्चलाई माथि पुश गरेर।

मानौं myproject एउटा प्रोजेक्ट हो जसमा फाइलहरू छन्:

# John को कम्प्युटरमा

$ cd myproject

$ git init

$ git add .

$ git commit -m 'Initial commit'

$ git remote add origin git@gitserver:/srv/git/project.git

$ git push origin master

यस बिन्दुमा, अरूले पनि रिपोजिटरीलाई सजिलै क्लोन गर्न र परिवर्तनहरू माथि पुश गर्न सक्छन्:

$ git clone git@gitserver:/srv/git/project.git

$ cd project

$ vim README

$ git commit -am 'Fix for README file'

$ git push origin master

यस विधिको प्रयोगले तपाईंले थोरै विकासकर्ताहरूका लागि छिट्टै read/write Git सर्भर सेटअप गर्न सक्नुहुन्छ। तर ध्यान दिनुपर्छ कि हाल यी सबै प्रयोगकर्ताहरूले git प्रयोगकर्ताका रूपमा सर्भरमा लग इन गरी shell पहुँच गर्न सक्छन्। यदि तपाईंले यो सीमित गर्न चाहनुहुन्छ भने, /etc/passwd फाइलमा shell परिवर्तन गर्नुपर्नेछ।

तपाईं git-shell नामको सीमित शेल टुल प्रयोग गरेर git प्रयोगकर्ता खाता केवल Git सम्बन्धी गतिविधिहरूसम्म सीमित गर्न सजिलै गर्न सक्नुहुन्छ।  
git-shell Git सँगै आउँछ।  
यदि तपाईंले यसलाई git प्रयोगकर्ता खाताको लग इन शेलको रूपमा सेट गर्नुभयो भने, त्यो खाताले सर्भरमा साधारण शेल पहुँच पाउन सक्दैन।  
यसका लागि, तपाईंले त्यस खाताको लग इन शेलको रूपमा bash वा csh सट्टा git-shell निर्दिष्ट गर्नुपर्छ।

**सेटअप प्रक्रिया:**

1. **जाँच गर्नुहोस् कि git-shell पहिले नै /etc/shells मा छ कि छैन:**

$ cat /etc/shells

यदि छैन भने, यो थप्नको लागि:

$ which git-shell # सुनिश्चित गर्नुहोस् कि git-shell तपाईंको सिस्टममा स्थापित छ।

$ sudo -e /etc/shells # र, git-shell को पथलाई पछिल्लो कमाण्डबाट `/etc/shells` मा थप्नुहोस्।

1. **git प्रयोगकर्ताको शेललाई git-shell मा परिवर्तन गर्नुहोस्:**

$ sudo chsh git -s $(which git-shell)

**अब git प्रयोगकर्ताले:**

* **SSH कनेक्शन प्रयोग गरेर Git रिपोजिटरीहरूमा पुश र पुल गर्न सक्छ।**
* **तर, सर्भरमा शेल लग इन गर्न सक्दैन।**  
  यदि कसैले प्रयास गर्छ भने, यस्तो लग इन अस्वीकृति देखिनेछ:

$ ssh git@gitserver

fatal: Interactive git shell is not enabled.

hint: ~/git-shell-commands should exist and have read and execute access.

Connection to gitserver closed.

**SSH पोर्ट फर्वार्डिङ रोक्ने तरिका:**

git सर्भरले कुनै अन्य होस्टलाई पहुँच दिन सक्षम हुन नदिन, तपाईं authorized\_keys फाइललाई सम्पादन गरेर प्रत्येक कुञ्जीको अगाडि यी विकल्पहरू थप्न सक्नुहुन्छ:

no-port-forwarding,no-X11-forwarding,no-agent-forwarding,no-pty

**परिणाम यसरी देखिनुपर्छ:**

$ cat ~/.ssh/authorized\_keys

no-port-forwarding,no-X11-forwarding,no-agent-forwarding,no-pty ssh-rsa

AAAAB3NzaC1yc2EAAAADAQABAAABAQCB007n/ww+ouN4gSLKssMxXnBOvf9LGt4LojG6rs6h

PB09j9R/T17/x4lhJA0F3FR1rP6kYBRsWj2aThGw6HXLm9/5zytK6Ztg3RPKK+4kYjh6541N

YsnEAZuXz0jTTyAUfrtU3Z5E003C4oxOj6H0rfIF1kKI9MAQLMdpGW1GYEIgS9EzSdfd8AcC

IicTDWbqLAcU4UpkaX8KyGlLwsNuuGztobF8m72ALC/nLF6JLtPofwFBlgc+myivO7TCUSBd

LQlgMVOFq1I2uPWQOkOWQAHukEOmfjy2jctxSDBQ220ymjaNsHT4kgtZg2AYYgPqdAv8JggJ

ICUvax2T9va5 gsg-keypair

no-port-forwarding,no-X11-forwarding,no-agent-forwarding,no-pty ssh-rsa

AAAAB3NzaC1yc2EAAAADAQABAAABAQDEwENNMomTboYI+LJieaAY16qiXiH3wuvENhBG...

अब **Git नेटवर्क कमाण्डहरू सामान्य रूपमा काम गर्छन्**, तर प्रयोगकर्ताहरू सर्भरमा शेल लग इन गर्न सक्दैनन्।

**थप अनुकूलन:**

* git प्रयोगकर्ताको होम डिरेक्टरीमा एउटा **विशेष डिरेक्टरी** बनाएर git-shell कमाण्डलाई केही अनुकूलन गर्न सकिन्छ।
* यसले सर्भरले स्वीकार गर्ने **Git कमाण्डहरूमा सीमितता राख्न** वा
* **SSH प्रयास गर्दा देखिने सन्देशलाई अनुकूलन गर्न** मद्दत गर्छ।

git-shell कमाण्डलाई अनुकूलन गर्न थप जानकारीको लागि:

$ git help shell

------------------------------------------ END -----------------------------------------