# **Mercurial over SSH**

Getting Continua to work with a mercurial repository over SSH takes a couple of additional steps.

# Step 1.) Install Putty

Download the set of putty tools at http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Install them into a directory on the machine your Continua server is running.

C:\putty					
Include in library 🔻 Share with 🔻 Burn New folder					
Name *	Date modified	Туре	Size		
B PAGEANT.EXE	10/07/2012 9:32 AM	Application	136 KB		
PLINK.EXE	10/07/2012 9:32 AM	Application	304 KB		
PSCP.EXE	10/07/2012 9:32 AM	Application	316 KB		
PSFTP.EXE	10/07/2012 9:32 AM	Application	324 KB		
😰 РИТТҮ.СНМ	10/07/2012 9:32 AM	Compiled HTML Help	439 KB		
PUTTY.CNT	10/07/2012 9:32 AM	CNT File	32 KB		
PUTTY.EXE	10/07/2012 9:32 AM	Application	484 KB		
PUTTY.HLP	10/07/2012 9:32 AM	Help file	643 KB		
PUTTYGEN.EXE	10/07/2012 9:32 AM	Application	176 KB		

## Step 2.) Generate a public/private key

In the putty directory you just installed, look for the executable called **PuTTYgen.exe**. Open it and click the **Generate** button. You'll need to move the mouse cursor around until the progress bar reaches the end.

le <u>K</u> ey Con <u>v</u> ersions <u>H</u> elp -Key Please generate some randomness by	moving the ma	ouse over the bla	ank area.
Key Please generate some randomness by r	moving the ma	ouse over the bla	ank area.
Please generate some randomness by	moving the mo	ouse over the bla	ank area.
Actions			
Generate a public/private key pair			<u>G</u> enerate
Load an existing private key file			Load
Save the generated key	58	ve p <u>u</u> blic key	Save private key
Parameters			
Parameters Type of key to generate:			
Parameters Type of key to generate: OSSH-1 (RSA) OSSH	1-2 <u>R</u> SA	C ss	:H-2 <u>D</u> SA

Do not set a passphrase on your private key. Continua CI does not currently support private keys with a passphrase.

Once it's finished, click the save private key button, select Yes when it asks you to save the key without a pass phrase.

PuTTY Key Generator		? ×					
<u>File K</u> ey Con <u>v</u> ersions <u>H</u> elp							
_ Key							
Public key for pasting into OpenS	Public key for pasting into OpenSSH authorized_keys file:						
ssh-rsa AAAAB3NzaC1yc2EAAAABJQA zaGHW6GThU1igvAMud +SeCeBfoyPMOZHKu8kPkMRB 3Qheh0OypabizWvmQic	ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQEAgOlfvZUAX0Mh3KwRz6AWeiAETmRFwcoAe/ zaGHW6GThU1igvAMud +SeCeBfoyPMOZHKu8kPkMRBxlT4EslW7baAlAnHDUxF/330ljP7qHEQSlzjZMVHkTN 3Qheh0OypabizWvmQic						
Key fingerprint: ssh-rsa 2	Key fingerprint: ssh-rsa 2048 b3:01:db:d4:28:e3:8a:5a:d0:39:84:8a:çe:00:0b:0d						
Key <u>comment</u> : PuTTYgen Wa	rning	×					
Key p <u>a</u> ssphrase C <u>o</u> nfirm passphi	you sure you want to save this key hout a passphrase to protect it?						
Actions							
Generate a pub	Yes No	Generate					
Load an existing private key file	Load an existing private key file Load						
Save the generated key	Save p <u>u</u> blic key	<u>S</u> ave private key					
Parameters							
Type of key to generate: C SSH- <u>1</u> (RSA)	SSH-2 <u>R</u> SA O SSI	H-2 <u>D</u> SA					
Number of <u>b</u> its in a generated ke	у:	2048					

Name the file and save it in your chosen directory. I saved mine as C:\private\_key.ppk.

The private key file needs to be available on the Continua CI Server. We recommend naming the key with a name that matches the use of the key as defined in the next section. For example, if you are using the key as a deployment key on bitbucket, then name it something like bitbucket\_deploy\_owner.ppk - this will make life easier when it comes to selecting the correct private key file to use on your repo.

## Step 3.) Storing public key on the SSH Server

If your repository resides on a hosting website like codebashq or bitbucket, you'll need to follow their guide on how to add a public key to your repository hosted there. Bitbucket have a guide here.

If your repository is hosted on a machine which you have SSH access to then you will need access to it through a username and password. For machines using OpenSSH or a common variant, the public key you generated above will need to be put in the authorized\_keys file. You can create (or find) that file in the .ssh directory of the home directory of the user you logged in, eg. /home/username/.ssh/authorized\_keys

The public key needs to be in a format recognized by your SSH server. If it's the common ssh server (OpenSSH) that comes with \*nix variants then the format you want is what PuTTYgen.exe outputs in the field "Public key for pasting into OpenSSH authorized\_keys file:"

<u>e</u>	PuTTY	Key Generat	or		<u>?</u> ×
<u>F</u> ile	<u>K</u> ey	Con <u>v</u> ersions	<u>H</u> elp		
Г	Key —				
	<u>P</u> ublic k	ey for pasting i	nto OpenSSH authorize	d_keys file:	
	3Qheh	00ypabizWvm(		a la ENIZCa DIIVELogona (a	
	9m7s7	voQdrwSJ7nXj)	y5dlnIYsCkkeV5Wj1Xv	Fc2JBZ4kaXM0Oq3S0	mvFnFl94JdNXSy
	qUVOI: wIQ==	3/JQlyfv7UaH) rsa-key-201209	(alXgct8P0dOld/SXleoF )10	GyaG5GmMZnZWf9XJ	I6A0OM8j0PkgPp ▼
	, Key f <u>i</u> ng	gerprint:	ssh-rsa 2048 b3:01:db	:d4:28:e3:8a:5a:d0:39:8	84:8a:ce:00:0b:0d
	Key <u>c</u> or	nment:	rsa-key-20120910		
	Key p <u>a</u> s	ssphrase:			
	C <u>o</u> nfirm	passphrase:			
Г	Actions				
	Generat	te a public/priv	ate key pair		<u>G</u> enerate
	Load ar	n existing privat	e key file		Load
	Save th	e generated ke	у	Save p <u>u</u> blic key	Save private key
Г	Paramet	ters			
	Type of O SSH	key to generat H- <u>1</u> (RSA)	e: SSH-2 <u>R</u> SA	O SSI	H-2 <u>D</u> SA
	Number	r of <u>b</u> its in a ger	erated key:		2048

Paste it into your authorized\_keys file and save it.

# Step 4.) Configure your Mercurial Repository in Continua CI

## **Private Key File**

Fill in the full path to your private key file. The file must be on the same machine as the Continua CI Server Service, and accessible by the Service User.

### **SSH Command**

Continua uses TortoisePlink.exe (included with the server installer) for ssh tunnelling. The SSH Command is the command passed to hg.exe when running over ssh. The default command is setup for using a private key :

#### SSH Command using private key

```
"#plink# -ssh -2 -batch -C -i ""#PrivateKeyFile#"""
```

If your remote repository uses username & password authentication over ssh rather than a key pair, change the SSH Command to this :

#### SSH Command using username & password

"#plink# -ssh -2 -batch -C -1 #UserName# -pw #Password#"

The SSH Command includes tokens that are replaced at runtime :

#PrivateKeyFile# - replaced with the Private Key File property. #UserName# - replaced with the UserName property. #Password# - replaced with the Password property. #plink# - replaced with the PLink version selected.

## **PLink Version**

This defaults to the server property Server.PLink.Default (TortoisePlink.exe) - you can add support for other versions of PLink by adding server properties with a namespace of "Server.Plink".