Linux ssh public-key authentication.

This exercise needs 2 machines: *vm1* and *vm2*

Both machines should have a user: *linuser*

The outcome: linuser is able to login from vm1 to vm2, using
ssh without password-authentication.

1. On vm1 and on vm2, create the user.

```
[root@vm1 ~]# useradd linuser
[root@vm1 ~]# echo 'l!nus3r' | passwd linuser --stdin

[root@vm2 ~]# useradd linuser
[root@vm2 ~]# echo 'l!nus3r' | passwd linuser –stdin
```

2. On vm2 check whether publickey authentication is enabled.
If it is not enabled, enable it and restart sshd.

```
[root@vm2 ~]# grep -i pubkey /etc/ssh/sshd_config
PubkeyAuthentication no

[root@vm2 ~]# sed -i 's/PubkeyAuthentication no/PubkeyAuthentication yes/' \
/etc/ssh/sshd_config
[root@vm2 ~]# systemctl restart sshd
```

3. Switch to linuser and create a key-pair.

Note: By default, the keys will be created in $HOME/.ssh.
Press enter when you are asked for a passphrase.

```
[root@vm1 ~]# su - linuser
[linuser@vm1 ~]$ ssh-keygen -t rsa

[root@vm2 ~]# su - linuser
[linuser@vm2 ~]$ ssh-keygen -t rsa
```

4. Try to login as linuser from vm1 to vm2.

```
[root@vm1 ~]# ssh linuser@vm2
linuser@vm2's password:
```

Note: you can login by typing your password.

5. From vm1 copy the linuser public-key to vm2:/tmp/vm1_linuser_pubkey.

```
[linuser@vm1 ~]$ scp .ssh/id_rsa.pub linuser@vm2:/tmp/vm1_linuser_pubkey
```

6. Add the public-key to the authorized_keys file in .ssh for both users.

```
[linuser@vm2 ~]$ cat /tmp/ vm1_linuser_pubkey >> ~/.ssh/authorized_keys
```

7. On vm2 set the correct permissions for the authorized_keys file.

```
[linuser@vm2 ~]$ chmod 400 ~/.ssh/authorized_keys
```

8. As linuser, login from vm1 to linuser on vm2.

```
[linuser@vm1 ~]$ ssh linuser@vm2
Last login: Thu Jan 18 15:35:36 2018
[linuser@vm2 ~]$
```