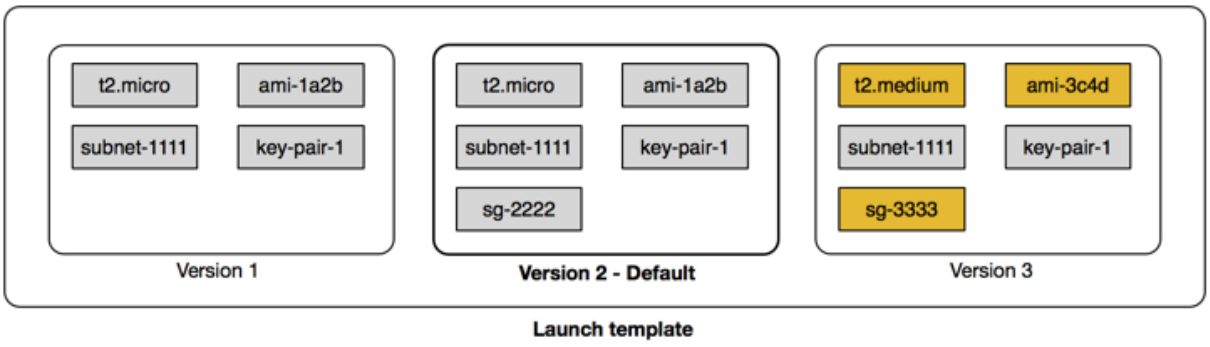
Launch an instance from a launch template

You can create a *launch template* that contains the configuration information to launch an instance. You can use launch templates to store launch parameters so that you do not have to specify them every time you launch an instance. For example, a launch template can contain the AMI ID, instance type, and network settings that you typically use to launch instances. When you launch an instance using the Amazon EC2 console, an AWS SDK, or a command line tool, you can specify the launch template to use.

For each launch template, you can create one or more numbered *launch template versions*. Each version can have different launch parameters. When you launch an instance from a launch template, you can use any version of the launch template. If you do not specify a version, the default version is used. You can set any version of the launch template as the default version—by default, it's the first version of the launch template.

The following diagram shows a launch template with three versions. The first version specifies the instance type, AMI ID, subnet, and key pair to use to launch the instance. The second version is based on the first version and also specifies a security group for the instance. The third version uses different values for some of the parameters. Version 2 is set as the default version. If you launched an instance from this launch template, the launch parameters from version 2 would be used if no other version were specified.



Re-using EC2 SSH Key Pair in multiple AWS regions

One of the parameters required for launching an EC2 instance is a *Key Pair* which is effectively an *SSH Key* used for interactive logging into the default user account — on Amazon Linux it's the *ec2-user* account — or for decrypting the Windows Administrator's password.

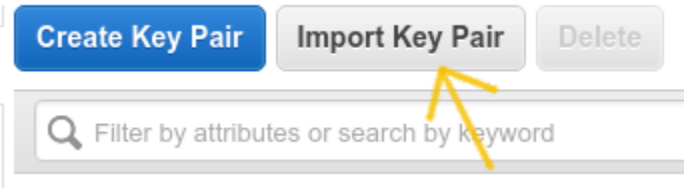
It is easy to create a new *Key Pair / SSH Key* as part of the EC2 launch process however as soon as you start using more regions and more

accounts you will quickly end up with heaps different Keys and unless you are diligent with their naming both on the filesystem and in AWS you will end up with a mess. Like I used to.

Fortunately there is a way to **re-use an existing Key Pair in other regions** or even in other AWS accounts. And it's actually pretty easy.

Importing existing Key Pair

To import this key to a new region switch to that region and go to *Services* → *EC2* → *Key Pairs* and click **Import Key Pair**.

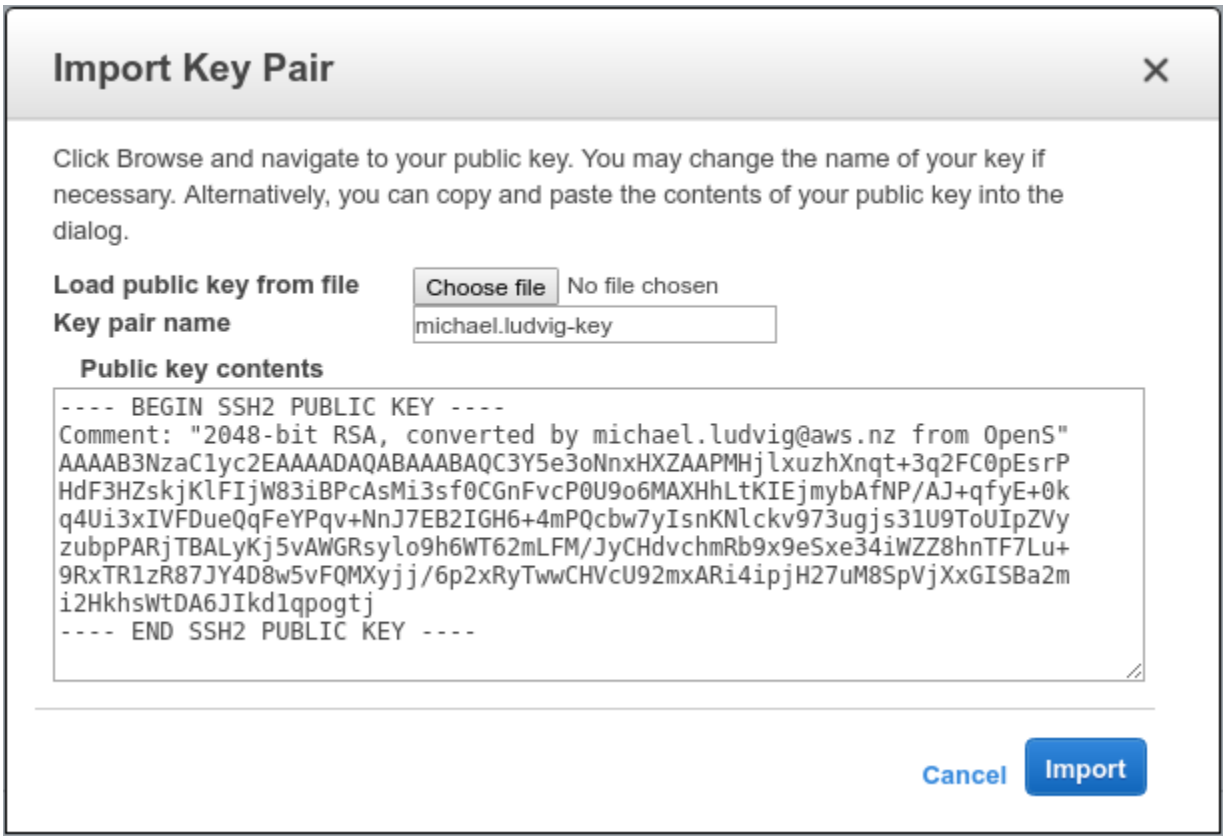


The key pair *name* must be unique within the region (i.e. you can't have two different keys with the same name) but you should use the same name in all the regions. Keeping it consistent across the regions greatly simplifies your automation — you won't need a per-region key name mapping.

The format of the public key must be in RFC4716 format, *not* in the openssh format starting with `ssh-rsa AAAAB3...` format that's suggested

by Amazon's docs yet rejected by the import tool. Convert any of your SSH keys to RFC4716 with this command:

```
~ $ ssh-keygen -e -m RFC4716 -f ~/.ssh/michael.ludvig-key.pem
Enter passphrase: ****
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "2048-bit RSA, converted by michael.ludvig from Open"
AAAAB3NzaC1yc2EAAAADAQABAAABAQC3Y5e3oNnxHXZAAPMHjlxuzhXnqt+3q2
HdF3HZskjKlFIjW83iBPcAsMi3sf0CGnFvcP0U9o6MAXHhLtKIEjmybAfNP/AJ
[...]
u9RxTR1zR87JY4D8w5vFQMXyjj/6p2xRyTwwCHVcU92mxARi4ipjH27uM8SpVj
a2mi2HkhsWtDA6JIkd1qpogtj
---- END SSH2 PUBLIC KEY ----
```



Using the same command you can also convert your public keys (e.g. `~/.ssh/id_rsa.pub`) to RFC4716 format and import them to Amazon.

Now simply copy and paste the output including the BEGIN end END lines and click Import.