RStudio Server (Pro) with Amazon EFS

More RStudio users are taking advantage of the cheap computation power of cloud servers. One challenge of using ephemeral compute instances (Amazon's EC2) is persisting data, files, folders beyond the lifetime of an EC2 instance.

Solution:
Instead of using an instance's local storage you can make use of Amazon's Elastic File System. EFS allows for permanent storage that is also highly available and optimized for I/O.

Step 1: Launch an Instance
Be sure to use a security group that allows outbound traffic:
http://docs.aws.amazon.com/efs/latest/ug/accessing-fs-create-security-groups.html

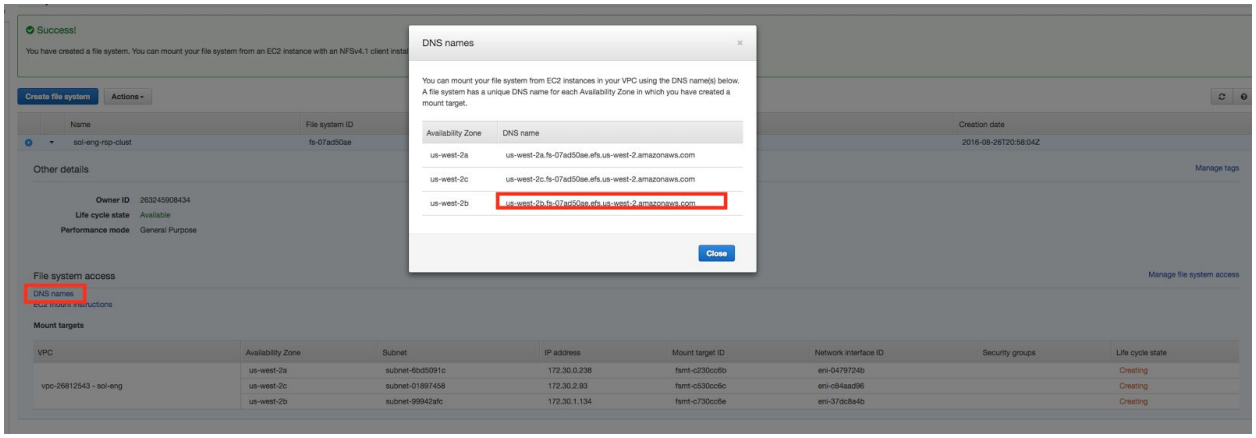
Step 2: Update R location / Install R / Install RSP
(The following code is for an Ubuntu 14.04 machine, but follow the relevant download instructions)

```
# update source to point to the latest version of R
sudo sh -c 'echo "deb http://cran.rstudio.com/bin/linux/ubuntu
trusty/" >> /etc/apt/sources.list'
gpg --keyserver keyserver.ubuntu.com --recv-key E084DAB9
gpg -a --export E084DAB9 | sudo apt-key add -

# get updates
sudo apt-get update

# install R
sudo apt-get install r-base
sudo apt-get install gdebi-core

# install RSP (you could also use the open source RStudio .deb)
sudo wget
https://download2.rstudio.org/rstudio-server-pro-0.99.903-amd64.deb
sudo gdebi rstudio-server-pro-0.99.903-amd64.deb

# activate RStudio Server Pro license (optional)
sudo rstudio-server license manager activate <product-key>
```

Step 3: Create Elastic File System
Be sure to use a security group that allows inbound nfs traffic from your ec2 security group
http://docs.aws.amazon.com/efs/latest/ug/accessing-fs-create-security-groups.html

Step 4: Mount File System within Instance

Get DNS Name from Elastic File System (referenced as **mount-target-DNS**)
http://docs.aws.amazon.com/efs/latest/ug/mounting-fs-mount-cmd-dns-name.html




```
sudo apt-get install nfs-common
sudo mount -t nfs4 -o vers=4.1  mount-target-DNS:/ /home
```

Set Up Auto-mounting (Optional):
http://docs.aws.amazon.com/efs/latest/ug/mount-fs-auto-mount-onreboot.html

Add the following line to the /etc/fstab file:
mount-target-DNS:/ /home nfs defaults,vers=4.1 0 0

Mount and Set-Up All In One (Optional):

```
sudo echo 'mount-target-DNS:/ /home nfs defaults,vers=4.1 0 0' >>
/etc/fstab
sudo mount -a
```

These steps are recommended as they allow you to pause and restart AWS instances without re-mounting the drives by hand.

Step 5: Add Users

```
sudo adduser test_user
```

The mount will pick up the correct file permissions automatically, to check:

```
ls -la /home
drwxr-xr-x  3 root      root      4096 Sep  2 22:43 .
drwxr-xr-x 22 root      root      4096 Sep  2 22:35 ..
drwxr-xr-x  2 test_user test_user 4096 Sep  2 22:43 test_user
```