More Auto-Backup Materials

Authentication and logging

The Reason for keychain

- For an **rsync** command to work, it must be possible to authenticate the user on remote machine.
- However, if you are using a **cron** task, you will not be there to type in a password or passphrase.
  - If you are using **SSH** keys, you could simply forego using a passphrase, altogether.
  - But, that is not ideal because it means your private key is *more vulnerable*.

The Reason for keychain

- One alternative you might consider would be to run **ssh-agent** and **ssh-add** prior to the **cron** job running.
  - This will not work, however, because your script will not have access to the environment variables needed by the **ssh** utility in order to use the agent.
  - Those variables are called **SSH_AGENT_PID** and **SSH_AUTH_SOCK**.
    - Basically, they tell the **ssh** utility where to look for your agent.
    - Even if your agent is running, **ssh** needs the variables to use it.

The Reason for keychain

- However, your script is running in a subshell that the **cron** daemon creates -- with only the basic environment variables.

- The **keychain** utility helps you overcome this.
  - Upon first logging into your interactive session, you can execute the following command:

```
keychain $HOME/.ssh/YOUR_PRIVATE_KEY
```

  - **YOUR_PRIVATE_KEY** may be **id_dsa**, **id_rsa**, etc. – depending on what you chose.

The Reason for keychain

- The command will...
  - Create an agent for you
  - Prompt you for your passphrase
  - Create some files in your **.keychain** directory
- Each of those files will be of the form

YOUR_HOSTNAME-SHELL

  - **YOUR_HOSTNAME** is the name of your machine
  - **SHELL** is a shell abbreviation
  - For example: **itvm29-4c-sh**

The Reason for keychain

- Each of those files will contain shell code that sets and exports the variables **SSH_AGENT_PID** and **SSH_AUTH_SOCK**.
- **Example:**

```
SSH_AUTH_SOCK=/tmp/ssh-9uCpNpvqdFjc/agent.1275; export SSH_AUTH_SOCK;
SSH_AGENT_PID=1276; export SSH_AGENT_PID;
```

- In **autobackup.sh**, just before the **rsync** command, add the following line:

```
source $HOME/.keychain/`/bin/hostname`-sh
```

The Reason for keychain

- This way, your script will execute the file and create the variables, enabling **ssh** to find and use your agent, *even though you are not interacting at present*.

- Sometimes, it may help to check and make sure your keychain information is still valid and consistent, which you can do in three steps:
  - Look at the file contents:

```
cat $HOME/.keychain/`hostname`-sh
```

The Reason for keychain

- Make sure the **ssh-agent** id matches:

```
ps aux | grep ssh-agent
```

- Make sure the **agent.XXXX** file exists:

```
ls /tmp | grep "ssh-"
```

- If there is an issue, then you may need to run the **keychain** utility again.

Other Project 9 Tips

- In the file **autobackup.sh**, you might replace this part

```
-e ssh
```

  with this

```
-e "ssh -v"
```

  for more verbose **ssh** output.
- The **-azvv** options for **rsync** make your *rsync* output verbose, but you might want more information from the **ssh** utility, as well.

Other Project 9 Tips

- When running **autobackup.sh** as a **cron** task, you might consider *redirecting* **stdout** and **stderr** into a log file.
  - That way, you can read the output for troubleshooting purposes
  - You can do that like so:

```
/usr/local/bin/autobackup.sh OTHER_VIRTUAL_MACHINE
      /guests/VIRTUAL_MACHINE_NAME > $HOME/cron.log 2>&1
```

  - This creates a file in your home directory called **cron.log**, which will contain the output from running **autobackup.sh**
  - This part **2>&1** redirects standard error into standard output, so that you get the contents of both.