Logging into the DCC

As a shared computing resource, all work done on the DCC must be done by submitting jobs to the scheduler. Login nodes must not be used to execute computing tasks.

Acceptable use of login nodes include:

- lightweight file transfers,
- script and configuration file editing,
- job submission and monitoring,
- BaseSpace BaseMount - users can use basemount to mount BaseSpace into their home directories.

To minimize disruption and ensure a comfortable working environment for users, resource limits are enforced on login nodes, and processes started there will automatically be terminated if their resource usage (including CPU time, memory and run time) exceed those limits.

Web Access

Users who are not comfortable using the command line can access the DCC through Open OnDemand, which provides a access to interactive sessions on the DCC through your browser. The Open OnDemand service. provides acesss to the DCC through RStudio, Jupyter, or Linux desktop sessions.

SSH Access

Login to the DCC using `ssh netid@dcc-login.oit.duke.edu`, (note: VPN is not required, but MFA **is required**).

```
kk338@CDSS-5630 ~ % ssh kk338@dcc-login.oit.duke.edu
Password:
Duo two-factor login for kk338

Enter a passcode or select one of the following options:

1. Duo Push to XXX-XXX-4007
2. Phone call to XXX-XXX-4007
3. Phone call to XXX-XXX-9784
4. SMS passcodes to XXX-XXX-4007 (next code starts with: 2)

Passcode or option (1-4): 1
Success. Logging you in...
Last login: Tue Dec 21 11:07:41 2021 from xxx
################################################################################
# MOTD                                                                         #
# My patch window is wednesday 03:00                                           #
################################################################################
kk338@dcc-login-03  ~ $
```

A word on ssh clients

Linux and macOS systems generally come with a SSH client already installed and you can `ssh` directly from a terminal window. For Windows users you will need an ssh client such as PuTTY or MobaXterm. Advanced users may consider installing WSL2 (Windows Subsystem for Linux 2), see installation instructions here, but beware DCC support staff do not

offer support for your local computer and software installations. Other ssh clients may work or not, but beware of clients that create persistent sessions to the DCC.

Users may connect to the DCC login nodes using VSCode to troubleshoot code and manage scripts. However, **VSCode SHOULD NOT BE USED** for computationally intense tasks on login nodes or on compute nodes via common or scavenger partitions. If you would like to execute code using VSCode, connect to OnDemand and run your code interactively using the CodeServer option.

SSH Keys

Setting up ssh keys from your workstation will greatly simplify the login and file transfer process by creating a secure key based session between your workstation and the DCC instead of using your password with MFA.

Step 1. To generate a key pair run the following (on your local machine) from your command shell prompt:

```
ssh-keygen -t ed25519
```

You should get prompted for what file to save the key in. The default should be fine (just hit enter).

Next you should be prompted for a passphrase. Pick something secure that you can remember.

The public key generated will be stored in the file:

```
~/.ssh/id_ed25519.pub
```

Step 2. View and copy your ssh key:

```
cat ~/.ssh/id_ed25519.pub
```

Please be sure that you do not accidentally leave off the .pub. The file ~/.ssh/id_ed25519 (without the .pub) is your private key file, which you should NEVER share with ANYONE.

Note that the contents of this file (what is displayed by cat) should look something like this:

ssh-ed25519 AAUAC3NzaC1lZDI1NTE5AAAAIDhkOqqB+kRDIDp86EGo0hNvogOxwJmM5kxx/aYCzOrG kk338@CDSS-CPP0H34NX2

Step 3. Add your public key to your Duke profile by copying to your “SSH Public keys” under “Advanced User Options” at: idms-web.oit.duke.edu/portal.

Find the section titled "Manage Your Public SSH Keys." In that section there should be clickable text that says "+ See More about SSH keys". Click that to expand this section, and you should see a textbox for "New Public Key" where you can paste your Public Key (copied in the previous step).

Step 4. Test your connection. Note `Enter passphrase for key` should display instead of `Password`

```
kk338@CDSS-5630 ~ % ssh kk338@dcc-login.oit.duke.edu
Enter passphrase for key '/Users/kk338/.ssh/id_ed25519':
 Last login: Tue Dec 21 11:13:41 2021 from xxx
###############################################################################
# MOTD                                                                        #
# My patch window is wednesday 03:00                                          #
###############################################################################
kk338@dcc-login-03  ~ $
```

Key Management

An "authentication agent" for SSH keys is typically referred to as "ssh-agent" which is a built-in tool on most operating systems that stores your SSH private keys in memory, allowing you to authenticate without re-entering your passphrase

every time you connect to a server; while a "password manager" like 1Password helps with management of passwords as well as SSH keys. **It will likely makes sense for you use either a password manager or ssh-agent, but not both**:

1Password (password manager)

- To get started:1Password
- Key Management: 1Password Manage Keys

ssh-agent (Linux)

Most modern Linux desktop environments will automatically prompt for and manage your ssh private key passphrase, if not,run the following: `ssh-add ~/.ssh/id_ed25519`

ssh-agent (Mac)

macOS 12.0 or newer, run:

```
ssh-add --apple-use-keychain ~/.ssh/id_ed25519
```

macOS older than 12.0, run:

```
ssh-add -K ~/.ssh/id_ed25519
```

(and then type the password you just used) which will add add the key to the authentication agent, and also save the password in your keychain.

Then edit `~/.ssh/config` and add these lines to it:

```
Host *
UseKeychain yes
AddKeysToAgent yes
IdentityFile ~/.ssh/id_ed25519
```

ssh-agent (Windows):

- Documentation Windows ssh-agent

Running the following works for Windows, as well: `ssh-add ~/.ssh/id_ed25519`

View more general information about ssh public key authentication at ssh.com.

VSCode

Users may connect to the DCC login nodes using VSCode to troubleshoot code and manage scripts. However, **VSCode SHOULD NOT BE USED** for computationally intense tasks on login nodes or on compute nodes via common or scavenger partitions. If you would like to execute code using VSCode, connect to OnDemand and run your code interactively using the CodeServer option or follow the below instructions for connecting to an interactive session using VSCode.

Before starting this process, make sure you have generated SSH keys for the DCC. See the SSH Keys section above. Using MFA via Duo Mobile will not work.

Step 1: SSH Config

In order to access the compute nodes, you'll first need to create an SSH config file. Navigate to where you store your SSH keys (by default on macOS, it's ~/.ssh) and create a text file named 'config'.

```
(base) rm145@CDSS-LGQFWT4QR4 ~ % cd ~
(base) rm145@CDSS-LGQFWT4QR4 ~ % cd ~/.ssh
(base) rm145@CDSS-LGQFWT4QR4 .ssh % touch config
```

Within this file, you'll want to add the following:

```
# SSH for DCC login node
Host dcc-login
  IdentityFile /path/to/privateKey
  HostName dcc-login.oit.duke.edu
  User netID

# SSH for DCC compute node
Host dcc-compute
  IdentityFile /path/to/privateKey
  HostName dcc-core-[x]
  User netID
  ProxyJump dcc-login
```

You'll need to replace the following with your information:

- IdentityFile path
- User netID
- HostName core number for the compute node; you won't know this until you create the interactive application job

Save the file.

Step 2: Create Interactive Job

You can do this using Open OnDemand. Once created, check the coompute node that you're connected to; this will be the compute node that you need to use in your SSH config file. Edit your config file under `HostName` for the dcc-compute.

Step 3: Connect with VSCode Now, we'll use VSCode to connect directly to the compute node reserved via the interactive job. You'll only have access to this compute node for the walltime you requested.

- Start VSCode and (macOS) hit command+shift+p, then select 'Remote-SSH: Connect to Host...'
- Select `dcc-login`
- You'll be prompted for the SSH key's passphrase; enter that and you should be connected to the DCC login nodes
- Go to 'File > New Window', then hit command+shift+p again and select 'Remote-SSH: Connect to Host...'
- This time, select `dcc-compute`
- You should be prompted to enter the SSH key passphrase again; do so, and you should be connected to the compute node