

Various bugs you may face and their solutions

1. Hanging at login

```
salloc: Pending job allocation 42979770
salloc: job 42979770 queued and waiting for resources
```

Each user is authorized to get only one resource allocation (see it as a computer). If you try to get another allocation, your request will hang.

There could be two possibilities:

- a. [Your session crashed and you do not have a terminal running → you need to “kill” the crashed session before you can get a new one:](#)

Running “*squeue -u username*” on the command prompt (to clarify: you do not need to type “”. Use only the text within.) will show the list of jobs you have running.

For example:

```
[levisse@jed2 ~]$ squeue -u levisse
JOBID PARTITION  NAME  USER ST  TIME  NODES NODELIST(REASON)
 42923003  standard interact levisse R   0:35   1 jst136
```

Then, you can kill the crashed session with “*scancel JOBID*”

For example:

```
[levisse@jed2 ~]$ scancel 42923003
[levisse@jed2 ~]$ squeue -u levisse

JOBID PARTITION  NAME  USER ST  TIME  NODES NODELIST(REASON)
```

At this point `run_edadk` will start working again.

- b. [You already have a session and you want to create several terminals to run parallel jobs](#)

We will follow the same approach, first, from a new terminal identify the node of your previously opened session using “*squeue -u username*”

```
[levisse@jed2 ~]$ squeue -u levisse
JOBID PARTITION  NAME  USER ST  TIME  NODES NODELIST(REASON)
 42923017  standard interact levisse R   0:09   1 jst004
```

Then, ssh to that node:

```
[levisse@jed2 ~]$ ssh jst004
```

Finally, setup the EDA environment in this terminal:

```
[levisse@jst004 ~]$ /work/fvlsi/rhel7.csh
```

```
Starting apptainer from
/work/fvlsi/edack/infra/apptainer/images/rhel7.20251016152804.sif...
Paths mounted:
- /work/fvlsi/edack/infra:/edack
- /work/fvlsi/edack/softs:/softs
- /work/fvlsi/edack/dkits:/dkits
- /work/fvlsi/edack/education:/education
- /work/fvlsi/shared_material_classes:/shared
- /tmp/tmp.g4rwNA6izE:/run
- /scratch
```

EDARHEL7[levisse@jst004 ~]\$ --> FROM HERE (i.e., WHEN YOU SEE “EDARHEL”), YOU CAN RUN YOUR COMMANDS.

2. Virtuoso lock files:

When using a cluster, the issue with virtuoso lock files happen more often as every session is done from a different computer. Cadence Virtuoso was originally designed as a collaborative tool. Several users could work in parallel on the same library at the same time. However, there is also a need to prevent users from editing the same cellview (what chaos would it be then!). In that sense, if a cellview is opened, virtuoso will create something called a “lockfile”. When a view is locked by virtuoso, it cannot be accessed by any other user.

If you open a view, and, for some reason, do not properly close virtuoso, or change server, or if you are unlucky (the tools sometimes crash). You sometimes cannot reopen your cellviews, with an error message about lock files.

Two solutions are possible:

- a. You could manually erase the lockfiles, which are generally located inside the design library with the “.cdslock” extension. This solution is dangerous as it involves using a rm command, and you may delete the wrong files.
- b. Use the cadence tool clsAdminTool (recommended solution).

```
> clsAdminTool
> ale yourlib
This command will list the lockfiles in the lib called “yourlib” (update it accordingly)
> are yourlib
This command will remove the lockfiles in the lib called “yourlib” (update it accordingly)
```