Kamiak Cheat Sheet

Logging in to Kamiak

ssh *your.name*@kamiak.wsu.edu
ssh -X *your.name*@kamiak.wsu.edu ***X11 forwarding***

Transferring Files to and from Kamiak

scp -r *myFile your.name*@kamiak.wsu.edu:~ ***Copy to Kamiak***
scp -r *your.name*@kamiak.wsu.edu:~/*myFile* . ***Copy from Kamiak***
rsync -avx *myFile your.name*@kamiak.wsu.edu:~ ***Synchronize***

Linux Commands

cd	***Go to home directory***
cd ..	***Go up one level (. is current directory)***
cd ~/*myPath*	***Go to path relative to home directory***
ls	***List members of current directory***
pwd	***Show path of current directory***
mkdir -pv *myFolder*	***Create a directory***
cp -r *myFrom myTo*	***Copy file, -r for entire directory***
mv *myFrom myTo*	***Move file***
rm -r *myFile*	***Delete file, -r for entire directory***
more *myFile*	***Display text file, one page at a time***
cat *myFile*	***Display entire text file***
man cp	***Manual page for command***
Ctl-c	***Kill current command***
Ctl-z	***Suspend current command***
bg	***Run suspended command in background***
fg	***Run background command in foreground***
disown -h	***Disconnect from terminal***

Text Editors

vi
nano
emacs

Submitting Batch Jobs to Kamiak

sbatch *myJob*.sh	***Submit a batch job script (to test, sbatch --test-only)***
squeue -u *your.name*	***View my pending and running jobs in the job queue***
squeue -j *jobNumber*	
scancel *jobNumber*	***Cancel a job***
sacct -S 2/26/18 -u *your.name*	***View job history including active jobs***
scontrol show job *jobNumber*	***View job details***

Viewing Information about the Cluster

sinfo -a | more	***What partitions and nodes are available***
squeue -a | more	***View all running and queued jobs***
scontrol show node cn93	***View node details (memory, cpus, GPUs)***

Interactive Session on Compute Node

```
idev -N 1 --ntasks-per-node=2 --mem-per-cpu=8G -t 360   #SBATCH same options
module load python
python -i
        print "Hello World!"
        exit()
srun -l python helloWorld.py
exit
```

Using Available Software on Kamiak

module avail	***Available modules compatible with compiler***
module spider	***See all modules***
module load python	***Load latest version***
module load python3/3.5.0	***Load specific version***
module list	***See loaded modules***
module whatis anaconda3	***See what a module does***
module help wrf	***See help for a module***
module unload python3	***Unload a module***
module purge	***Unload all modules***
which python	***See that python is in your path***
printenv PATH	***See effects of loading modules on environment***
printenv LD_LIBRARY_PATH	

Using Scratch Storage

```
export myScratch = "$(mkworkspace -q)"   Create scratch folder, lifetime 2 weeks
export myScratch = "$(mkworkspace -q -b /local)"  Create scratch on /local SSD
```

Sample Job Script

```
#!/bin/bash
#SBATCH --partition=kamiak          # Partition/Queue to use
#SBATCH --job-name=myJob            # Job name
#SBATCH --output=myJob_%j.out       # Output file (stdout)
#SBATCH --error=myJob_%j.err        # Error file (stderr)
#SBATCH --mail-type=ALL             # Email notification: BEGIN,END,FAIL,ALL
#SBATCH --mail-user=your.name@wsu.edu  # Email address for notifications
#SBATCH --time=7-00:00:00           # Wall clock time limit Days-HH:MM:SS

#SBATCH –-nodes=1                   # Number of nodes (min-max)
#SBATCH –-ntasks-per-node=2         # Number of tasks per node (max)
#SBATCH –-ntasks=2                  # Number of tasks (processes)
#SBATCH --cpus-per-task=1           # Number of cores per task (threads)

echo "I am job $SLURM_JOBID running on nodes $SLURM_JOB_NODELIST"
module load python                  # Load module from Kamiak repository
srun python helloWorld.py           # Each srun is a job step, spawns --ntasks
                                    # Each task runs this program (total 2 times)
echo "Completed job on node $HOSTNAME“
```

Sample Job Array

```
#!/bin/bash
#SBATCH --partition=kamiak          # Partition/Queue to use
#SBATCH --job-name=myJobArray       # Job name
#SBATCH --output=myJobArray_%A_%a.out       # Output filename
#SBATCH --error=myJobArray_%A_%a.err        # Error filename, group_index
#SBATCH --time=7-00:00:00           # Wall clock time limit Days-HH:MM:SS
#SBATCH --mail-type=ALL             # Email notification: BEGIN,END,FAIL,ALL
#SBATCH --mail-user=your.name@wsu.edu  # Email address for notifications
```

```
#SBATCH --array=0-2:1          # Number of jobs, in steps of 1
#SBATCH --nodes=1              # Number of nodes (min-max)
#SBATCH --ntasks-per-node=1    # Number of tasks per node (max)
#SBATCH --cpus-per-task=1      # Number of cores per task (threads)
#SBATCH --mem-per-cpu=8G       # Memory per core (gigabytes)

# Runs this job 3 times, with index SLURM_ARRAY_TASK_ID as 0,1,2
# Split your data into 3 files, name them array_0.txt, array_1.txt, array_2.txt
# Each job array step is scheduled as an individual job
# Each job array step is allocated the above resources (cores, memory)

echo "Starting job array $SLURM_ARRAY_TASK_ID on host $HOSTNAME"
module load python
srun python helloWorld.py "data/array_${SLURM_ARRAY_TASK_ID}.txt"
echo "Completed job array $SLURM_ARRAY_TASK_ID on host $HOSTNAME"
```

Other Types of Jobs

MPI message passing

```
#SBATCH –nodes=2                    Multiple nodes, tasks do not share memory
#SBATCH –ntasks-per-node=2
#SBATCH –cpus-per-task=1
```

OpenMP shared memory

```
#SBATCH –nodes=1                    Single node, threads share memory
#SBATCH –ntasks-per-node=1
#SBATCH –cpus-per-task=20
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
```

GPU (Graphics Processing Unit) accelerators

```
#SBATCH –nodes=1
#SBATCH –ntasks-per-node=1
#SBATCH –cpus-per-task=1
#SBATCH –gres=gpu:tesla:1           Offload kernel functions to GPU
```

Getting Help

hpc.wsu.edu — ***Support & Drop-in Hours***