HPSS Deployment Automation

Livermore Computing

Alicja Gornicka & Gabriel Rusk
HPCCEA

August 2018

LLNL-PRES-XXXXXX

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

HPSS Team

Alicja Gornicka
University of California, Santa Cruz

Gabriel Rusk
Westminster College

Objectives

I. Use a configuration management system (i.e. ansible) to integrate the configuration of an HPSS test environment

II. Use the configuration management with the cluster deployment to fully automate the HPSS install

III. Enable efficient testing of the HPSS environment

IV. Present packaged automation to the HPSS team

Our Project

- Decide on which automation configuration management system to use
- Automate instructions from the HPSS deployment guide which do not require the GUI
- Automate the configuration of a test HPC cluster using ansible
- Introduce and teach ansible and its use to the HPC Academy members

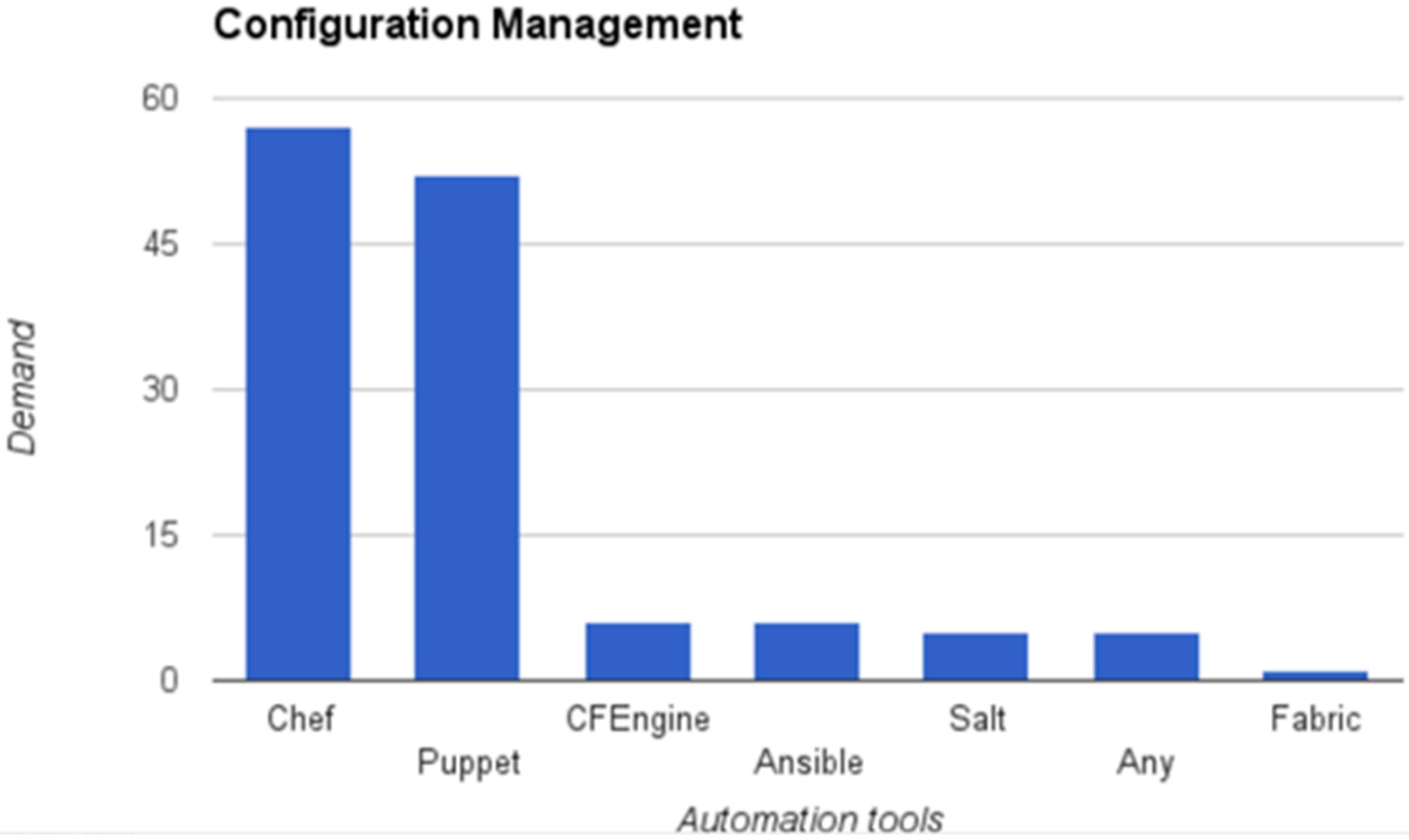
HPSS

High Performance Storage System

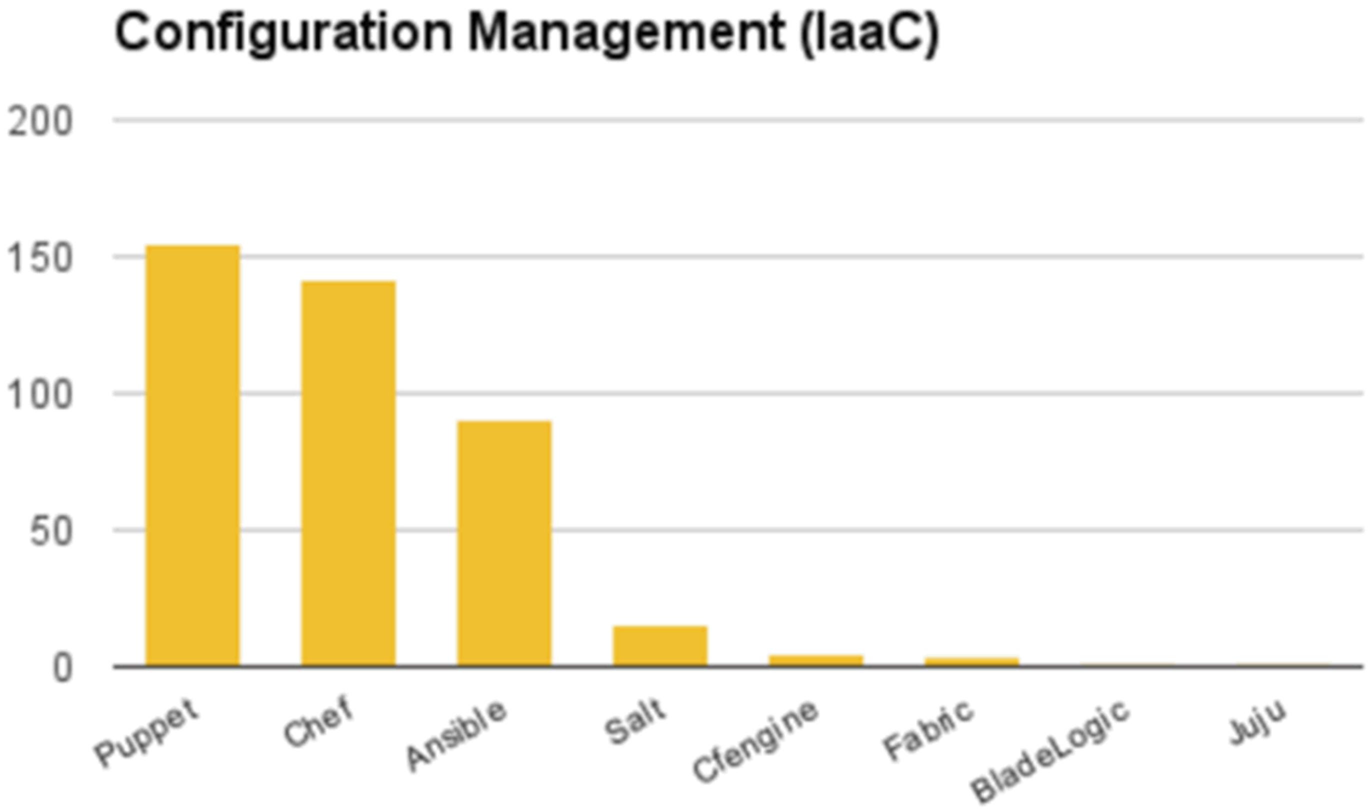
Ansible

- What is ansible?
  - Ansible is an open source configuration management tool that is quickly gaining popularity
  - Created to orchestrate multi-tier applications across clouds
  - Helps with software provisioning, configuration management, and application deployment
- Why ansible?
  - Created to counter difficulties of the other programs
  - The ansible team wanted to make something simple and easy to use so that people could just get and go
  - Well documented and has a great support community
  - All functions are performed over SSH

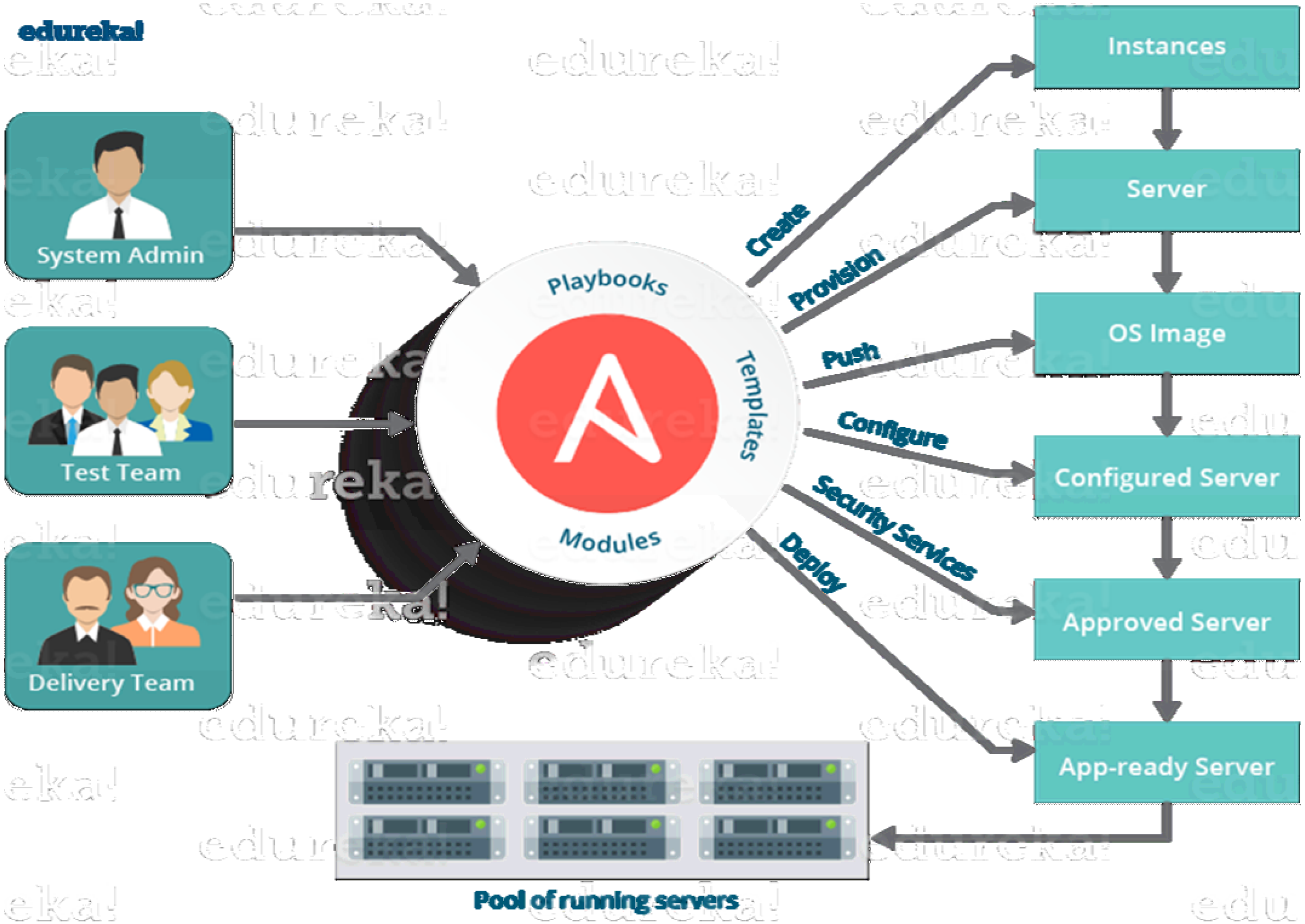
Ansible Usage in 2016



Ansible Usage in 2017



Ansible: Start to Finish



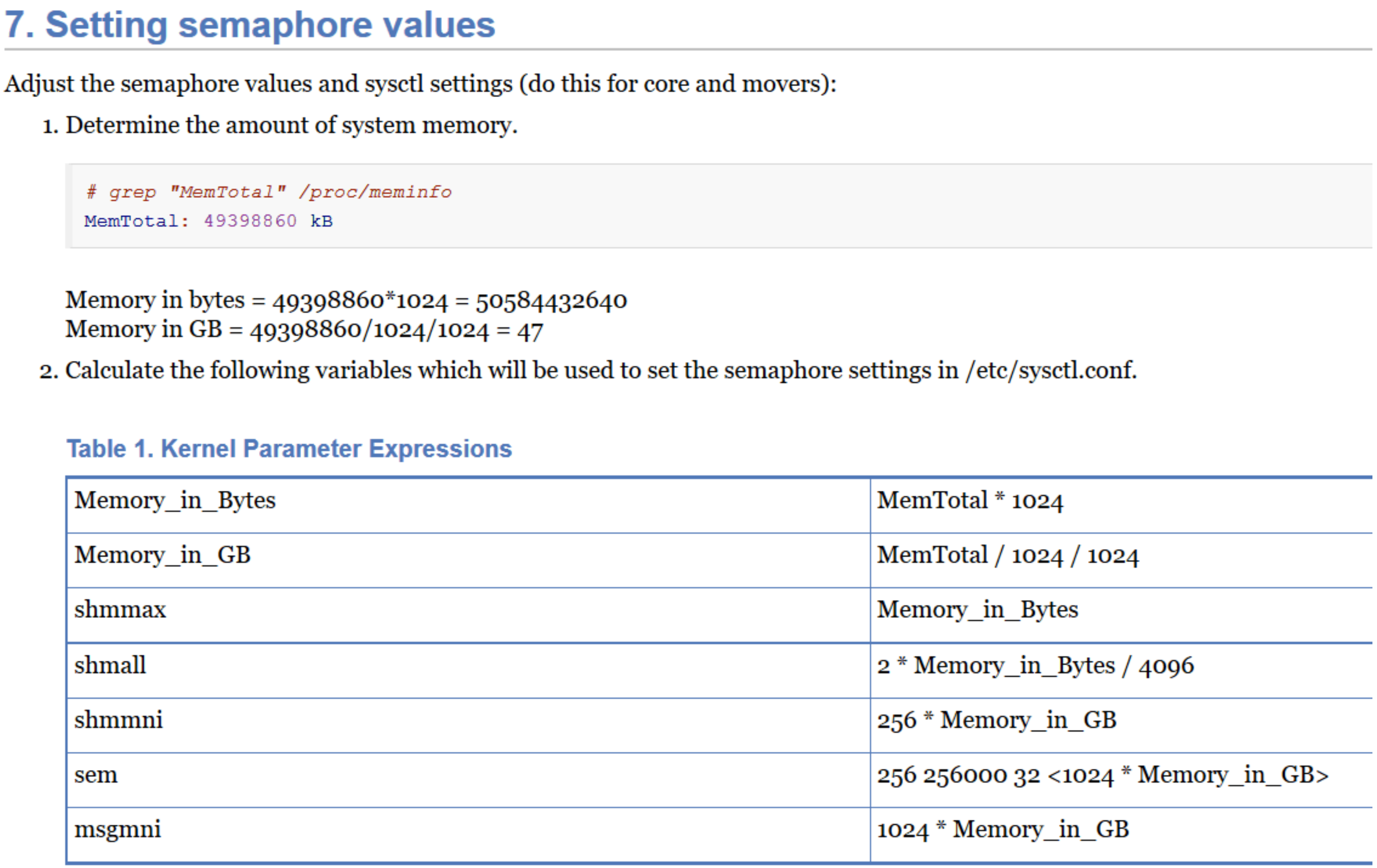
HPSS

- HPSS (High Performance Storage System) is a hierarchical file system software system designed to store and manage petabytes of data on disk and tape libraries in a network-centered, cluster-based environment
- HPSS is used in many large HPC sites (such as LLNL) to manage files on tape and disk

What we did

- Researched and studied what ansible is and how it works
- Implemented configuration tasks using ansible on the HPC test cluster
- Worked from an internal HPSS deployment document
  - Analyzed each section and split up the tasks
  - Automated most of sections 6-13 using ansible
    - Some sections could not be automated due to use of a GUI
    - Other sections were optional or used tools not required by LLNL
  - Used the HPC test cluster to test the automated HPSS deployment

The Automation of Section 7



7. Setting semaphore values

Adjust the semaphore values and sysctl settings (do this for core and movers):

1. Determine the amount of system memory.

```
# grep "MemTotal" /proc/meminfo
MemTotal: 49398860 kB
```

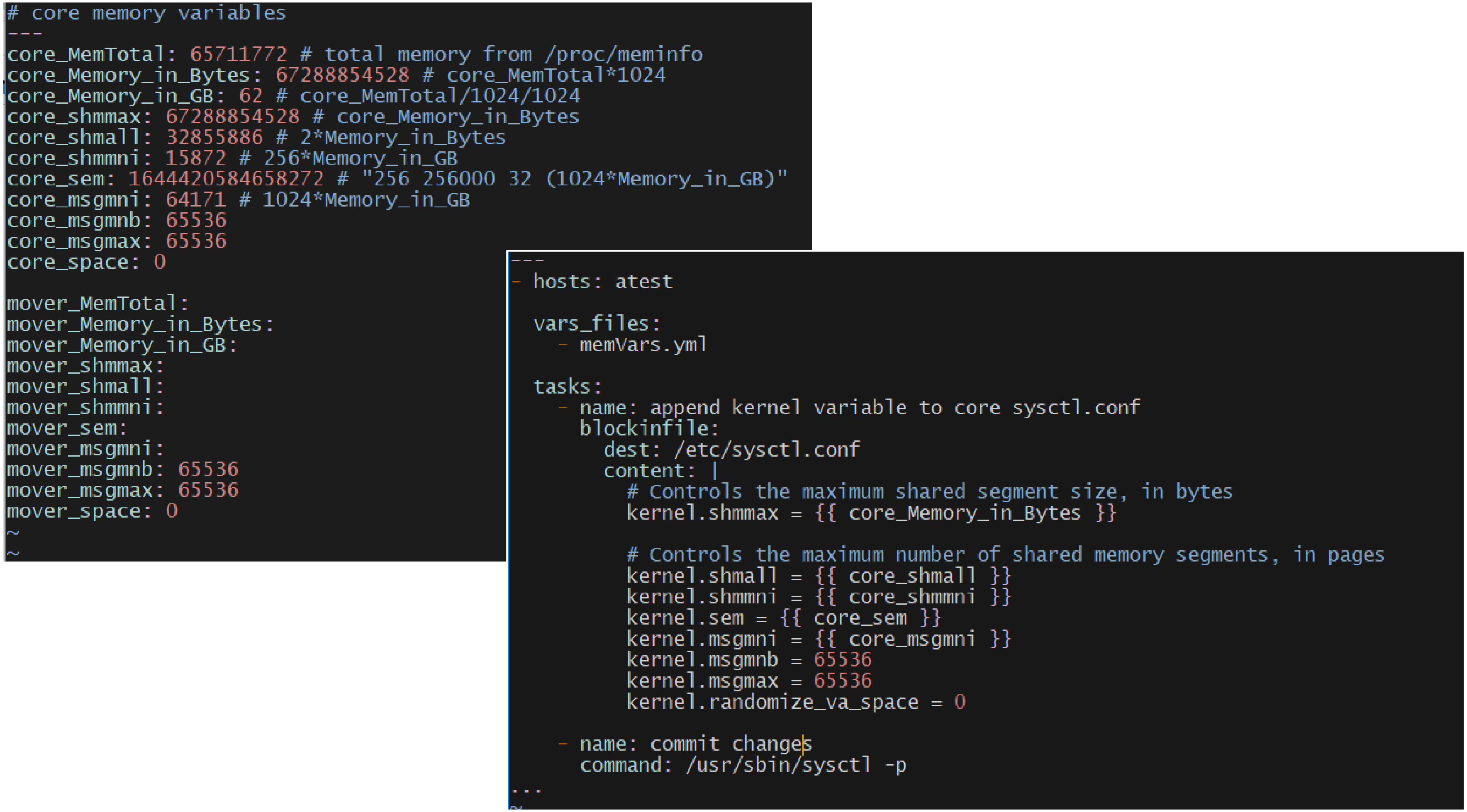
Memory in bytes = 49398860*1024 = 50584432640
Memory in GB = 49398860/1024/1024 = 47

2. Calculate the following variables which will be used to set the semaphore settings in /etc/sysctl.conf.

Table 1. Kernel Parameter Expressions

Memory_in_Bytes	MemTotal * 1024
Memory_in_GB	MemTotal / 1024 / 1024
shmmax	Memory_in_Bytes
shmall	2 * Memory_in_Bytes / 4096
shmmni	256 * Memory_in_GB
sem	256 256000 32 <1024 * Memory_in_GB>
msgmni	1024 * Memory_in_GB

The Automation of Section 7



```
# core memory variables
---
core_MemTotal: 65711772 # total memory from /proc/meminfo
core_Memory_in_Bytes: 67288854528 # core_MemTotal*1024
core_Memory_in_GB: 62 # core_MemTotal/1024/1024
core_shmmax: 67288854528 # core_Memory_in_Bytes
core_shmall: 32855886 # 2*Memory_in_Bytes
core_shmmni: 15872 # 256*Memory_in_GB
core_sem: 1644420584658272 # "256 256000 32 (1024*Memory_in_GB)"
core_msgmni: 64171 # 1024*Memory_in_GB
core_msgmnb: 65536
core_msgmax: 65536
core_space: 0

mover_MemTotal:
mover_Memory_in_Bytes:
mover_Memory_in_GB:
mover_shmmax:
mover_shmall:
mover_shmmni:
mover_sem:
mover_msgmni:
mover_msgmnb: 65536
mover_msgmax: 65536
mover_space: 0
~
~
```

```
---
- hosts: atest

  vars_files:
    - memVars.yml

  tasks:
    - name: append kernel variable to core sysctl.conf
      blockinfile:
        dest: /etc/sysctl.conf
        content: |
          # Controls the maximum shared segment size, in bytes
          kernel.shmmax = {{ core_Memory_in_Bytes }}

          # Controls the maximum number of shared memory segments, in pages
          kernel.shmall = {{ core_shmall }}
          kernel.shmmni = {{ core_shmmni }}
          kernel.sem = {{ core_sem }}
          kernel.msgmni = {{ core_msgmni }}
          kernel.msgmnb = 65536
          kernel.msgmax = 65536
          kernel.randomize_va_space = 0

    - name: commit changes
      command: /usr/sbin/sysctl -p
...
~
```

Challenges

- Getting the HPSS deployment documentation
- Time (haven’t verified end to end deployment process)
- Understanding ansible, ansible syntax, and ansible error handling
- Bringing the HPC test cluster back up after a power glitch
- Documentation targeted someone familiar with HPSS and had access to the standard testing environment
- Wanted to use ansible modules rather than just using the ansible command line option
  - Figuring out how to do it the “ansible way”

Conclusion

- Ansible
  - Easy to use and quick to pick up
  - Must be very careful with syntax (tabs vs spaces)
  - Allows flexibility for what we wanted to do (HPSS and cluster build)
  - Good fit for automating the HPSS documentation
- Successfully automated required sections of the HPSS deployment document
  - On track to achieving goal of minimal human command line input
  - Once required variables are saved into the variable files, running one command installs the specified section file

What's Next?

- Continue automation of the HPSS deployment
  - Clean up commenting and code
  - End to end testing and verification process
  - Adding in flags to allow user to specify which part they want to install
- Create a detailed README
  - How to use ansible to automate parts of the HPSS deployment documentation
  - Include what prerequisites are needed before running playbook
- Use ansible to automate the cluster install of the HPC test cluster
  - Fully automate the cluster build and package it for future reference
  - Make it easy to add in minor changes

Roll Credits

Thank you to **Jean, Dave, Thomas, Bryan, Rigo, Phil, Mike, and Geoff**

HPC Academy <3

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.