Los Alamos
NATIONAL LABORATORY
EST.1943

Delivering science and technology
to protect our nation
and promote world stability

How to Automate and not Manage under Rhine/Redwood

CUG'16

London, UK

Paul Peltz Jr, Daryl Grunau,
Adam DeConinck
HPC Group

5/12/2016

NNSA National Nuclear Security Administration
Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

How to not Drown in the Rhine

CUG'16
London, UK

Paul Peltz Jr, Daryl Grunau,
Adam DeConinck
HPC Group

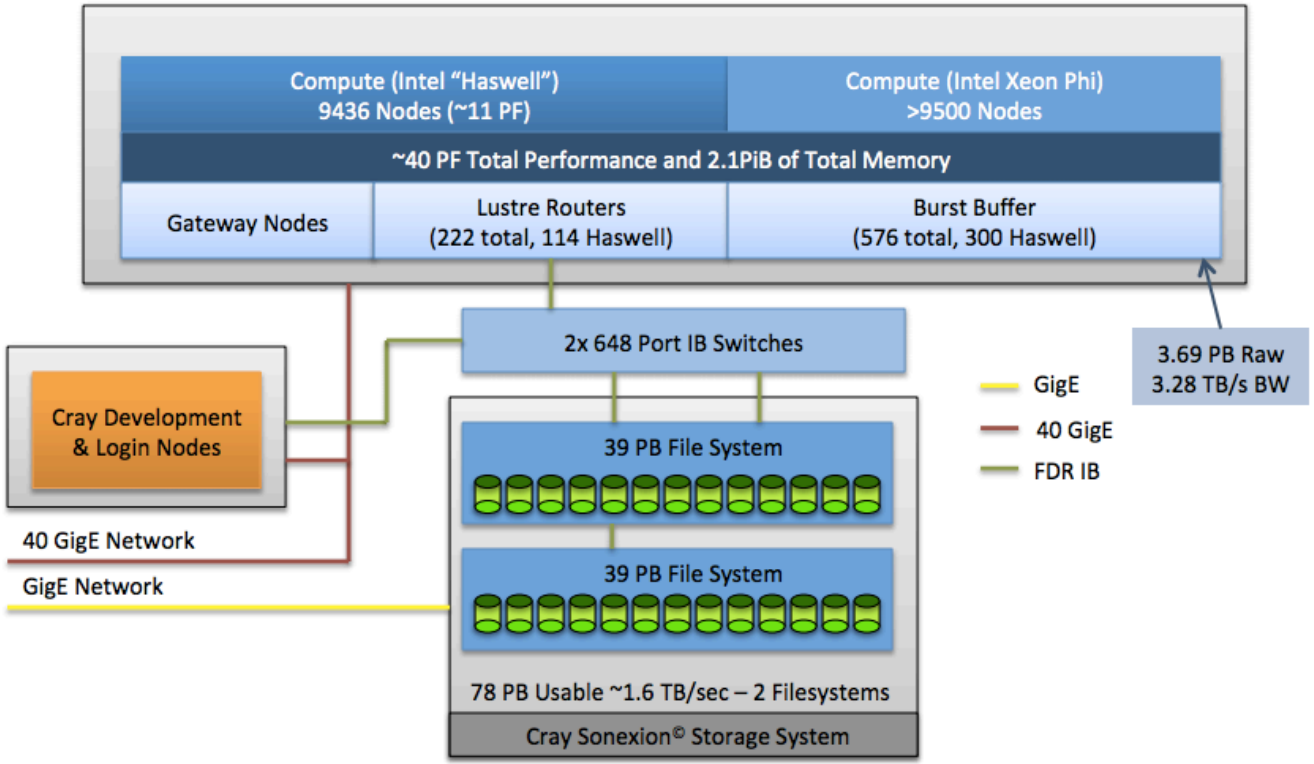
5/12/2016

NNSA National Nuclear Security Administration
Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

Presentation Overview

- **Trinity Architecture**
- CLE 6.0/SMW 8.0
- Cray Philosophy
- LANL Philosophy
- Configuration Management Challenges
- Revision Control
- Programming Environment
- Rolling Updates and Staged Upgrades
- Conclusions

Trinity Architecture



Presentation Overview

- **Trinity Architecture**
- **CLE 6.0/SMW 8.0**
- **Cray Philosophy**
- **LANL Philosophy**
- **Configuration Management Challenges**
- **Revision Control**
- **Programming Environment**
- **Rolling Updates and Staged Upgrades**
- **Conclusions**

CLE 6.0/SMW 8.0 (Rhine/Redwood)

- **Why did LANL upgrade to Rhine/Redwood?**
  - Risk Mitigation
  - Drive and Improve the development of the release

CLE 6.0/SMW 8.0 (Rhine/Redwood)

- **Why did LANL upgrade to Rhine/Redwood?**
  - Risk Mitigation
  - Drive and Improve the development of the release
- **What is new in this release?**
  - Spoiler Alert…everything!
  - Well…everything but the low level XT commands, ALPS, RUR, etc.
  - Shared root is gone
  - Leverages modern Linux and configuration management tools
    - RPM/Zypper
    - Ansible

Presentation Overview

- Trinity Architecture
- CLE 6.0/SMW 8.0
- Cray Philosophy
- LANL Philosophy
- Configuration Management Challenges
- Revision Control
- Programming Environment
- Rolling Updates and Staged Upgrades
- Conclusions

System Management Philosophy

Cray's Philosophy

- **Images not shared root**
  - Service/Compute/Login/eLogin
- **Configuration Set**
  - YAML files
  - Source of truth for system configuration
- **Ansible configuration management**
  - Nodes boot generic image
  - Ansible specializes node

System Management Philosophy
Cray's Philosophy (cont.)

- **Image Management and Provisioning Service (IMPS)**
  - Defines, creates, and exports images
    - Repos
    - Package collections
- **Node Image Mapping Service (NIMS)**
  - Image->node mapping for the boot process
  - config sets
  - boot parameters

System Management Philosophy

Cray's Philosophy (cont.)

- **ESMS/CIMS/CSMS/CMC**
  - Replaces Bright
  - OpenStack
  - SMW exports images to glance
- **eLogin**
  - Uses same repos as the internal system
  - Same PE image

Presentation Overview

- Trinity Architecture
- CLE 6.0/SMW 8.0
- Cray Philosophy
- LANL Philosophy
- Configuration Management Challenges
- Revision Control
- Programming Environment
- Rolling Updates and Staged Upgrades
- Conclusions

System Management Philosophy

LANL's Philosophy

- **Use of the Cray Ansible Area**
  - /etc/init.d/cray-ansible
    - Runs full playbook
    - Heavy-weight
    - only runs at boot
- **When to use the Site Ansible Area**
  - p0/ansible
  - Only what is necessary for the system to be booted fully configured
    - Errors can break boot process
    - Every play is evaluated

System Management Philosophy
LANL's Philosophy (cont.)

- **Local Configuration Management**
  - For single node use, SMW for example
  - Configuration preservation
  - Can be run periodically with less of a system impact
  - Compute nodes
    - Job interference
    - Resource Manager prologue/epilogue scripts
  - Service nodes
    - Full Ansible run less of an impact
    - Still only run a small subset of plays to minimize system impact

System Management Philosophy
LANL's Philosophy (cont.)

- **File Distribution**
  - Shared root gone
  - Requires script, Ansible play, or parallel distributed copy command
  - Leverage IMPS Distribution Service (IDS)
    - 9p file system shared throughout the system
    - /var/opt/cray/imps-distribution
  - Possible Issues
    - If service can reference/include file in IDS space
    - Symbolic link if supported
    - If service requires a reload or restart then other considerations will still need to be made

System Management Philosophy
LANL's Philosophy (cont.)

- Using Images and Recipes Effectively
  - Repos
    - Production Repo
    - Development Repo
  - Package Collections
    - Group like packages together (Work Load Manager)
    - Do not inherit a package collection within another
  - Change Tracking

```
"compute_lanl": {
"description": "Compute packages",
"package_collections": {},
"packages": { "bash-completion": { "rationale": "RT#1142443" },
"vtune": { "rationale": "RT#1123901" }}
```

Presentation Overview

- Trinity Architecture
- CLE 6.0/SMW 8.0
- Cray Philosophy
- LANL Philosophy
- Configuration Management Challenges
- Revision Control
- Programming Environment
- Rolling Updates and Staged Upgrades
- Conclusions

Configuration Management Challenges

Image Management

- **Live node image is modifiable**
  - tmpfs
  - Easy to quickly affect package changes to the system (Security fixes)
- **Modify image root before it is packaged**
  - Place files in the image root
- **Downsides to these approaches**
  - System skew
  - Reproducibility
  - imgbuilder builds from recipe not previous image

Configuration Management Challenges

Affecting an Image before it is packaged

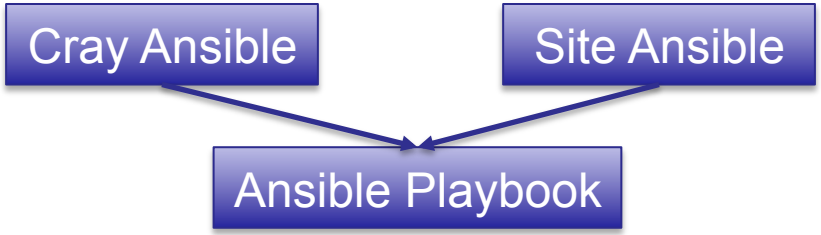
- Cray provides hooks into the image build process

```
"example_image_recipe": {
    "package_collections": {},
    "packages": {},
    "postbuild_copy": [
            "/home/crayadm/post_conf.sh",
            "/home/crayadm/post_conf_files/" ],
    "postbuild_chroot": [
            "${IMPS_POSTBUILD_FILES}/post_conf.sh],
    "repositories": {} },
```

System Management Philosophy

Fine Grained Control of the Ansible Playbook

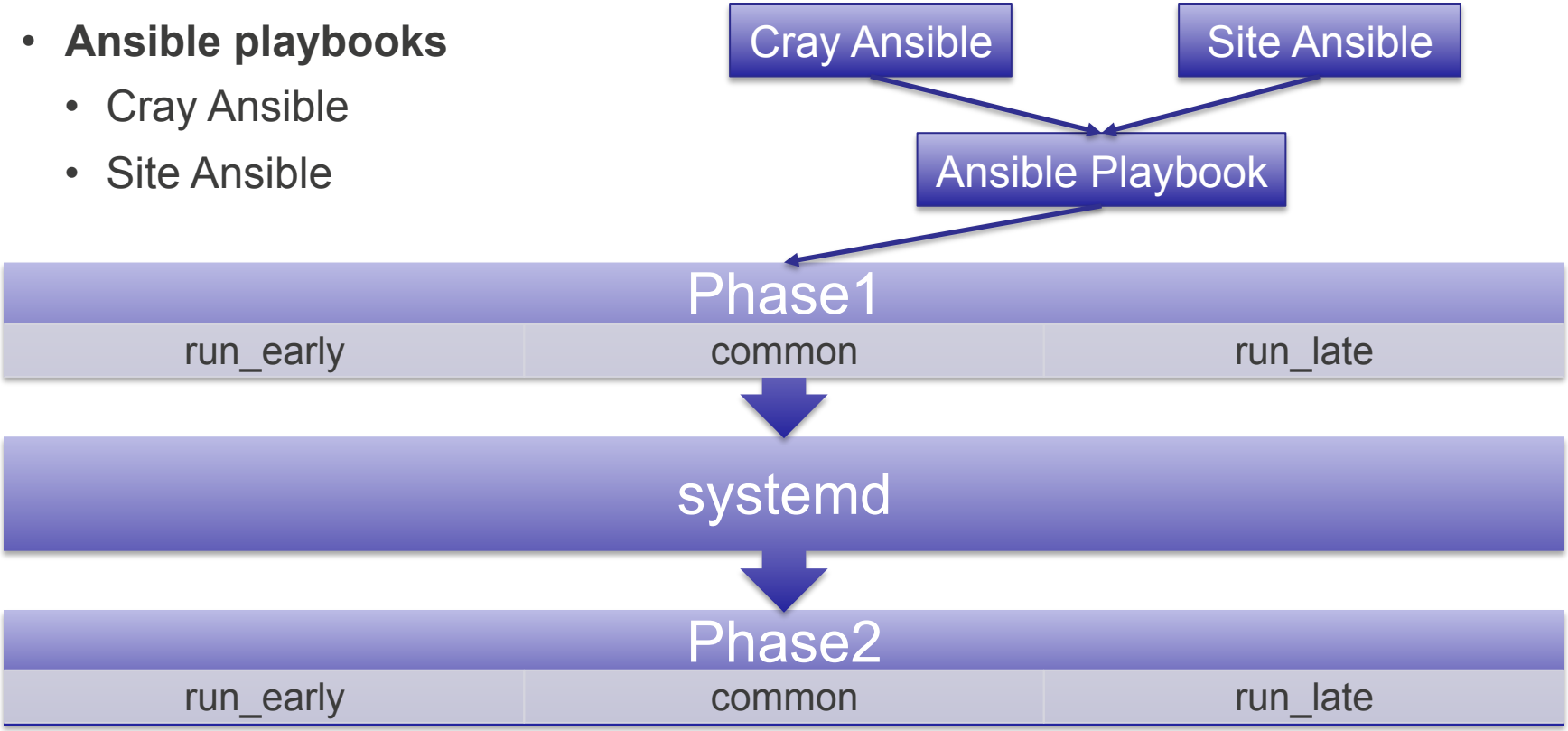
- **Ansible playbooks**
  - Cray Ansible
  - Site Ansible



System Management Philosophy

Fine Grained Control of the Ansible Playbook

- **Ansible playbooks**
  - Cray Ansible
  - Site Ansible



Configuration Management Challenges

Fine Grained Control of the Ansible Playbook (cont.)

- **Ansible variables affect ordering**
  - run early – a set of plays, first run stage (Boolean)
  - run late – a set of plays, last run stage (Boolean)
  - run after – run after specific play (Multival)
  - run before – run before specific play (Multival)

Configuration Management Challenges

Fine Grained Control of the Ansible Playbook (cont.)

- **Sample Ansible play**

```
---
- hosts: localhost
  vars:
    run_early: true
    run_after:
      - persistent_data
    run_before:
      - llm
    roles: - syslog
```

Configuration Management Challenges

Fine Grained Control of the Ansible Playbook (cont.)

- **Affecting Phase Selection**
  - When in_init
  - Absence of evaluation will run in both phases
- **Sample Ansible play**

```
---
# Run only when not in init
- include: example.yaml
      when: ansible_local.cray_system is defined
      and not ansible_local.cray_system.in_init
```

Configuration Management Challenges

Fine Grained Control of the Ansible Playbook (cont.)

Useful conditional statements (ansible –m setup localhost)

- All compute nodes
  - ansible_local.cray_system.platform == ”compute”
- All service nodes
  - ansible_local.cray_system.platform == ”service”
- All DataWarp nodes
  - ‘nvme0n1’ in ansible_devices
- All internal login nodes
  - ansible_local.cray_system.hostid in cray_login.settings.login_nodes.data.members
- All eLogin nodes
  - ansible_local.cray_system elogin is defined
- The SDB node(s)
  - ’sdb’ in ansible_local.cray_system.roles
- The SMW node(s)
  - ’smw’ in ansible_local.cray_system.roles
- The boot node(s)
  - ’boot’ in ansible_local.cray_system.roles

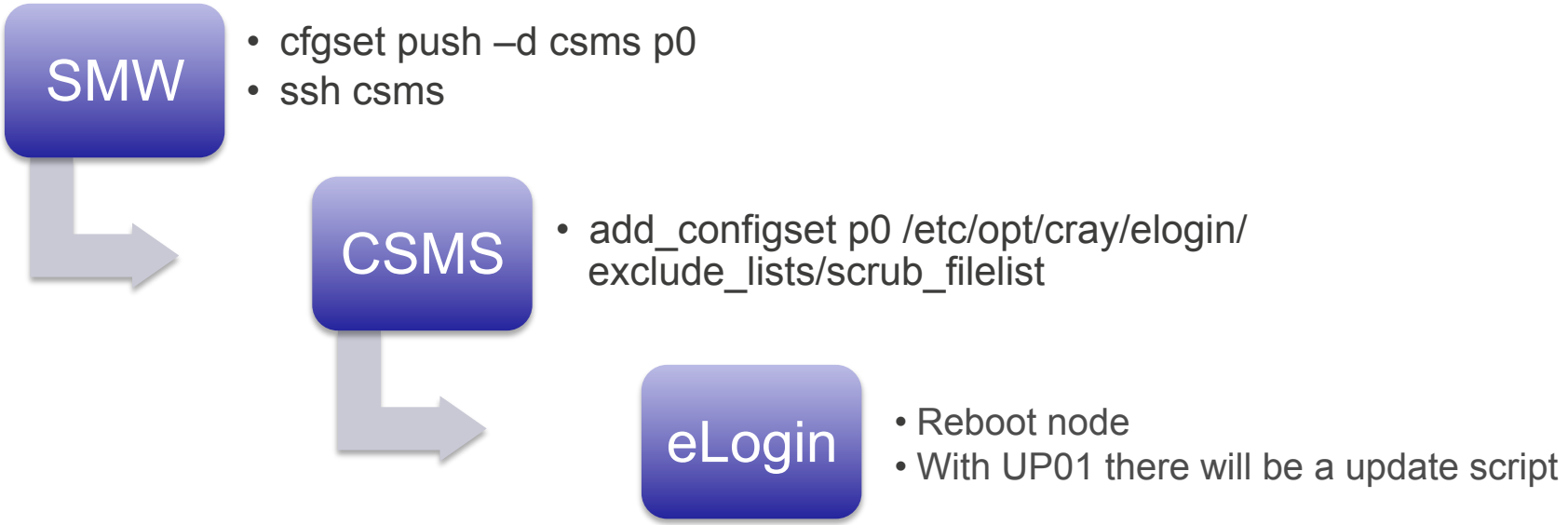
Configuration Management Challenges

Simple Sync vs. Configuration Management

- Simple Sync
  - Cray provided play that can place files on
    - Node classes
      - Common/Compute/Service/SDB
    - Node cnames
    - Node hostnames
  - Issues
    - Starts early but only useful for non configuration files
    - Service daemon requirement

Configuration Management Challenges

eLogin Challenges



Presentation Overview

- Trinity Architecture
- CLE 6.0/SMW 8.0
- Cray Philosophy
- LANL Philosophy
- Configuration Management Challenges
- Revision Control
- Programming Environment
- Rolling Updates and Staged Upgrades
- Conclusions

Revision Control of System Configuration

Objectives

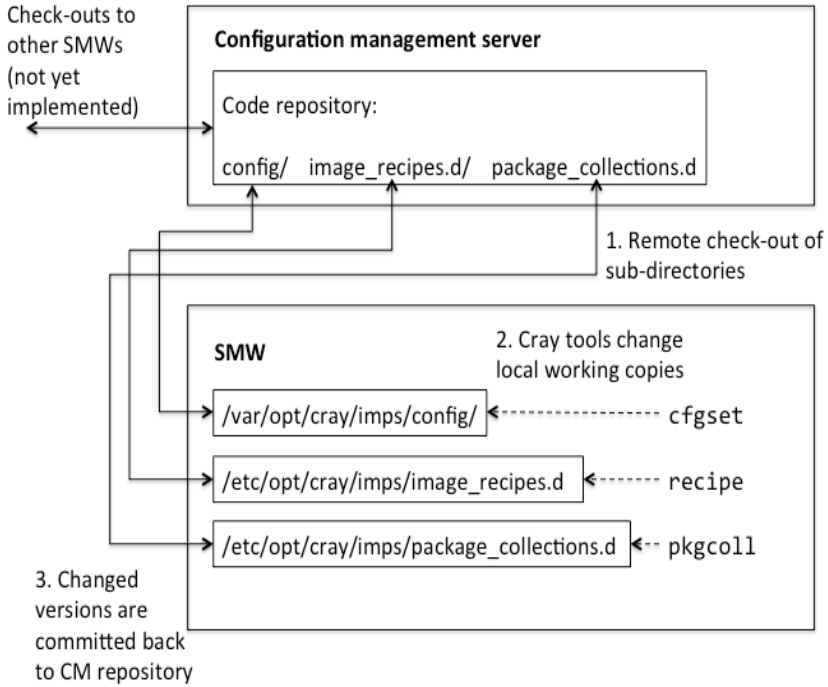
1) Keep a record of all changes to configuration files and management scripts, from initial installation to decommissioning
2) Attach commentary to each set of changes (e.g., “reconfigured network settings according to ticket 912“)
3) Associate each change with the particular person
4) View the differences between two revisions of the system configuration
5) Roll the configuration back to any previous revision
6) Share common files and scripts between multiple HPC systems

Revision Control of System Configuration

Insufficiency of Built-In Tools for Change Management/ Revision Control Implementation

- Backups are not sufficient
  - p0-autosave-$DATE
  - these get rolled off and auto deleted after time when cfgset is executed
- Changelogs are not sufficient

- Three areas under revision control
  - Config Set (p0)
  - Image Recipes
  - Package Collections



Revision Control of System Configuration

Challenges for our approach

- File Ownership and Permissions Within the Config Set
  - Certain directories and files need permissions or ownership set
  - The Cray Ansible plays do not always set this when copying the files to the node
- Change Management Challenges on the SMW
  - Root must be able to commit from the SMW to the svn repo
  - Cfgset, pkgcoll, recipe commands run as root and modify areas under rev control
- Requirement That Certain Changes be Made on the SMW
  - Feature request to test and validate changes before promoting them to the SMW

Presentation Overview

- Trinity Architecture
- CLE 6.0/SMW 8.0
- Cray Philosophy
- LANL Philosophy
- Configuration Management Challenges
- Revision Control
- Programming Environment
- Rolling Updates and Staged Upgrades
- Conclusions

Programming Environment Management

- Cloned Compute node image in UP00, no longer the case in UP01
- Just a directory in the image_roots with other system images
- Bind mounts specific directories
  - /opt/cray, /opt/gcc, /opt/java, /opt/intel, etc.
  - List is controlled by Cray and cannot be currently extended
- Future versions of the PE will allow more fine grained control of what is installed and removed from the PE
- There are specific files within the PE area that need to be under CM control
  - Default modules

Presentation Overview

- Trinity Architecture
- CLE 6.0/SMW 8.0
- Cray Philosophy
- LANL Philosophy
- Configuration Management Challenges
- Revision Control
- Programming Environment
- Rolling Updates and Staged Upgrades
- Conclusions

Rolling Updates and Staged Upgrades

- Rolling Updates (Compute Only)
  - Set aside development nodes
  - Using NIMS it is possible to test new software
    - Images
    - Config sets (including site Ansible plays)
    - PE images
- Staged Upgrades
  - UP01->UP02
  - Chroot into SMW BRTFS snapshot
  - Do Upgrade
  - Either reboot later into new chroot or use it as a test for install into the current snapshot

Presentation Overview

- Trinity Architecture
- CLE 6.0/SMW 8.0
- Cray Philosophy
- LANL Philosophy
- Configuration Management Challenges
- Revision Control
- Programming Environment
- Rolling Updates and Staged Upgrades
- Conclusions

Conclusions

- Use a TDS if at all possible!
- Not a simple system upgrade
- SMW and boot RAID will be reformatted
  - No going back unless there is a spare SMW and boot RAID
- Learning Ansible is essential to understanding booting and operation of machine
  - Especially if using the Site Ansible area
  - Learn enough to not break the boot process!
- Worksheets are your friend
  - Allows for pre-configuration of the system
  - Can be imported at install time

Questions?

Los Alamos
NATIONAL LABORATORY
EST.1943

Delivering science and technology
to protect our nation
and promote world stability