



# Ansible/Ascender Basics Networking



# Agenda

## MORNING SESSIONS - 9:00 a.m. - 12:00 p.m.

- **What is Ansible**
- **How Ansible Works and its Key Components**
- **Ascender Demo**
- **What is Ledger**
- **Hands-on**
  - Build my first playbook
  - Create a Job Template in Ascender
  - Execute my first playbook
  - Resource modules

## AFTERNOON SESSIONS - 12:00 p.m. - 3:30 p.m.

- **Hands-on**
  - Workflows
  - Variables and loops
  - Facts and conditionals
  - Surveys

# Agenda

COFFEE & PREP - 8:30 a.m. - 9:00 a.m. PST (11:30 -12pm ET)

MORNING SESSIONS - 9:00 a.m. - 12:00 p.m. PST (12pm - 3pm ET)

- What is Ansible
- How Ansible Works and its Key Components
- Ascender Demo
- What is Ledger
- Hands-on
  - Build my first playbook
  - Create a Job Template in Ascender
  - Execute my first playbook
  - Variables
  - Conditionals and blocks
  - Handlers
  - Loops

LUNCH - 12:00 p.m. - 1:00 p.m. (3pm- 4pm ET)

AFTERNOON SESSIONS - 1:00 p.m. - 3:30 p.m. (4pm - 6:30pm ET)

- Hands-on
  - Templates
  - Roles
  - Surveys
  - Workflows
  - Role Based Access Control(RBAC)
  - Wrap Up

# Housekeeping

- Team CIQ is here to help
- This is for you - please interact - **MUTE MICS**
  - Ask questions in chat during presentations
  - Ask questions out loud/in chat during labs
- Limit distractions - phone/email/twitch streams
- Take bio breaks during lab if you can so you don't miss instructions
- If you don't complete lab in time given I may extend or you can catch up during next lab
- Working lunch(I'll tag it together with a lab so also good for catch up)
- We will keep these labs up for 24 hours after the close of class, so you can complete/review later if desired.
- We need feedback(positive and constructive)
  - Please let us know what you liked or how we could improve



## 1.1 Ansible/Ascender/Ledger Overview



# What Is Ansible

Open-source automation tool for efficient infrastructure management

- Agentless
- YAML based automation scripts(playbooks)
- Cross-platform compatibility
- Scalable
- Active community

# Ansible Playbooks

```
---
- name: Perform package update
  hosts: web
  gather_facts: false
  tasks:
    - name: Update all packages on system
      ansible.builtin.dnf:
        name: "*"
        state: latest
    - name: Collect current linux version
      ansible.builtin.shell: cat /etc/os-release | grep PRETTY_NAME
      register: os_version_after
    - name: Display the OS version
      ansible.builtin.debug:
        var: os_version_after
```

# Ansible Module

An Ansible module is a reusable piece of code that is referenced in a playbook that performs a specific task.

- Generally written in python or powershell
- Majority are community or vendor provided
- Can create custom if desired

```
tasks:  
  - name: Update all packages on system  
    ansible.builtin.dnf:  
      name: "*"  
      state: latest
```



# Ansible Plugins

A component that extends Ansible functionality

- Lookup - retrieves data from an external resource
- Filter - allows for data manipulation(IE variable regex changes)
- Etc.

```
- name: Display the OS version  
ansible.builtin.debug:  
    var: os_version_after | to_nice_json
```

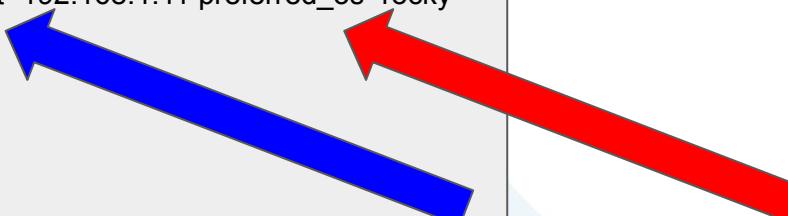


# Ansible Inventory

An inventory is a list of hosts that Ansible could possibly automate against. Below is a file based inventory.

- Can contain **magic** and **custom** variables

```
[web]
webserver1 ansible_host=192.168.1.10 preferred_os=rocky
webserver2 ansible_host=192.168.1.11 preferred_os=rocky
[database]
dbserver1
[loadbalancer]
lbserver1 ansible_user=ansible
```

A diagram illustrating how labels map to specific host definitions. A blue arrow points from the [database] label to the dbserver1 entry. A red arrow points from the [loadbalancer] label to the lbserver1 entry.

# Ansible Roles

An additional option for making portions of playbooks reusable

- Groups parts of playbooks that are reused frequently into a structure that can be referenced in a playbook
- Like a function or subroutine in a programming language

```
- name: Call user role  
  ansible.builtin.include_role:  
    name: standard_user_role
```



# Ansible Collections

Packaging mechanism to install Ansible resources(how to get modules)

- Tar file containing resources
  - Modules
  - Playbooks
  - Plugins
  - Docs
  - Etc.
- Ex: `cisco.ios` collection contains all needed modules to maintain Cisco IOS devices

# Ascender

A tech preview of the new Ascender Automation user interface can be found [here](#).

## Dashboard

4 Hosts    1 Failed hosts    3 Inventories    0 Inventory sync failures    3 Projects    0 Project sync failures

Job status    Recent Jobs    Recent Templates

Past month    All job types    All jobs

Job Runs

Date	Job Runs
6/17	1
6/18	2
6/19	0
6/20	0
6/21	0
6/22	0
6/23	2
6/24	0
6/25	1
6/26	0
6/27	24
6/28	1
6/29	0
6/30	0
7/1	0
7/2	0
7/3	1
7/4	0
7/5	1
7/6	0
7/7	2
7/8	0
7/9	1
7/10	0
7/11	1
7/12	0
7/13	0
7/14	0
7/15	0
7/16	0
7/17	0

# Ascender

Ascender is an open-source management tool for the Ansible automation engine

- Centralized management
- GUI and API interfaces
- Role-Based Access Control(RBAC) and User management
- Version control integration
- Inventory management
- Logging
- Surveys
- Workflows

# Ascender RBAC

The image displays two screenshots of the Ascender Automation web application interface.

**Left Screenshot (Access - Users):**

- Header:** ASCENDER AUTOMATION
- Left Sidebar:** Templates, Credentials, Projects, Inventories, Hosts; Access (highlighted), Organizations, Users (highlighted), Teams.
- Page Content:** Title: Users. Search bar (Email dropdown, search icon, Add button, Delete button). Table headers: Username, First Name, Last Name, Role. One row is listed: admin (System Administrator).

**Right Screenshot (Administration - Settings):**

- Header:** ASCENDER AUTOMATION
- Left Sidebar:** Access (highlighted), Organizations, Users, Teams; Administration (highlighted), Credential Types, Notifications, Management Jobs, Instance Groups, Instances, Applications, Execution Environments, Topology View; Settings.
- Page Content:** Title: Settings. Section: Authentication. Sub-section: Enable simplified login for your Ascender Automation applications. Options: Azure AD settings, GitHub settings, Google OAuth 2 settings, LDAP settings, RADIUS settings, SAML settings, TACACS+ settings, Generic OIDC settings.

# Ascender Inventories

The screenshot shows the Ascender Automation web application. The top navigation bar features the Ascender Automation logo. The left sidebar has two main sections: "Views" and "Resources". Under "Views", the "Inventories" option is selected and highlighted with a green bar at the bottom. Under "Resources", the "Inventories" option is also selected and highlighted with a green bar at the bottom. The main content area is titled "Inventories". It includes a search bar with a dropdown for "Name", a search icon, an "Add" button, and a "Delete" button. Below the search bar is a table header with "Name" and an upward arrow. Three rows are listed in the table:

- Demo Inventory
- Rocky LTS Demo
- SNOW Provision VM

This screenshot shows the "Hosts" tab for the "Rocky LTS Demo" inventory. The top navigation bar and sidebar are identical to the first screenshot. The main content area shows the "Hosts" section with the following table structure:

Name	Description	Related Groups
greg-rocky86lts		Its_demo
greg-rocky86nonlts		Its_demo

CiQ

# Ascender Projects

The screenshot displays the Ascender Automation web application interface across three main sections: a left sidebar, a top navigation bar, and a central content area.

**Left Sidebar:** Contains navigation links under 'Views' (Dashboard, Jobs, Schedules, Activity Stream, Workflow Approvals) and 'Resources' (Templates, Credentials, Projects, Inventories). The 'Projects' link is currently selected.

**Top Navigation Bar:** Features the Ascender Automation logo, a bell icon, a question mark icon, and a dropdown menu for the user 'admin'.

**Central Content Area:**

- Projects View:** Shows a list of projects with columns for Name and Status. Projects listed include Ascender VMWare, Demo Project, and Rocky LTS Mixed Demo, all marked as successful.
- Project Details View:** Shows detailed information for the 'Rocky LTS Mixed Demo' project. The 'Details' tab is selected. Key details include:
  - Last Job Status: Successful
  - Name: Rocky LTS Mixed Demo
  - Organization: Default
  - Source Control Type: Git
  - Source Control Revision: 569e4bb
  - Source Control URL: https://github.com/gsowell/ascender-rocky-lts
  - Cache Timeout: 0 Seconds
  - Project Base Path: /var/lib/awx/projects
  - Playbook Directory: \_8\_rocky\_lts\_mixed\_demo
  - Created: 6/16/2023, 5:07:50 PM by admin
  - Last Modified: 6/16/2023, 5:07:50 PM by admin

# Ascender Credentials

The screenshot shows the Ascender Automation web application interface. The top navigation bar includes the Ascender Automation logo, a menu icon, and a notification bell. The left sidebar has sections for Views (Dashboard, Jobs, Schedules, Activity Stream, Workflow Approvals), Resources (Templates, **Credentials**, Projects, Inventories, Hosts), and Access. The 'Credentials' section is currently selected. The main content area is titled 'Credentials' and displays a list of credentials with columns for Name and Type. The list includes:

Name	Type
Ansible Galaxy	Ansible Galaxy/Automation Hub API Token
Demo Credential	Machine
Lab Servers Root	Machine
Rocky LTS Mixed Demo Mnt Key	Mountain Access Key
VMWare Server	generic credential 1

# Ascender Job Templates

The image shows the Ascender Automation web application interface. On the left, there's a navigation sidebar with sections for Views (Dashboard, Jobs, Schedules, Activity Stream, Workflow Approvals) and Resources (Templates, Credentials, Projects, Inventories, Hosts). The 'Templates' section is currently selected. The main content area displays a list of job templates:

Name	Type
Demo Job Template	Job Template
Rocky LTS Mixed Demo	Job Template
SNOW Provision VM	Job Template
SNOW Rocky LTS Mixed Demo	Job Template

To the right, a detailed view of the 'Rocky LTS Mixed Demo' template is shown:

Details		Access	Notifications	Schedules
Name	Rocky LTS Mixed Demo	Job Type	run	
Inventory	Rocky LTS Demo	Project	Rocky LTS Mixed Demo	
Playbook	rocky-update-mixed-lts.yml	Forks	0	
Timeout	0	Show Changes	Off	
Created	6/16/2023, 5:08:57 PM by admin	Last Modified	6/16/2023, 5:13:08 PM by admin	
Credentials	SSH: Lab Server... Cloud: Rocky LT...			

# Ascender Job Output

The screenshot shows the Ascender Automation interface. The top navigation bar includes a logo, a search bar, and a user dropdown. The left sidebar has sections for Views (Dashboard, Jobs, Schedules, Activity Stream, Workflow Approvals), Resources (Templates, Credentials, Projects, Inventories, Hosts), and Access. The main content area is titled "Jobs" and displays a table of 82 jobs. The columns include Name, Status, Type, Start Time, and Finish Time. One job is expanded to show "Cleanup Job Details".

Back to Jobs		Details	Output		
Job ID	76	Status	<span>Successful</span>	Started	7/6/2023, 10:24:11 AM
Finished	7/6/2023, 10:37:39 AM	Job Template	Rocky LTS Mixed Demo	Job Type	Playbook Run
Launched By	admin	Inventory	Rocky LTS Demo	Project	Rocky LTS Mixed Demo
Revision	569e4bb61efadbe947b382b905c3aaa42bd0226a	Playbook	rocky-update-mixed-lts.yml	Verbosity	0 (Normal)
Execution Environment	AWX EE (latest)	Controller Node	ascender-task-78d449fb6-tsm6	Container Group	default
<pre>0 SSH password: 1 2 PLAY [Perform package update with LTS support] **** 3 4 TASK [Collect what version of linux is currently running] **** 5 ok: [greg-rocky86nonlts] 6 ok: [greg-rocky86lts] 7 8 TASK [Display the os_version for each server] **** 9 ok: [greg-rocky86lts] =&gt; { 10   "os_version_before.stdout": "PRETTY_NAME=\"Rocky Linux 8.6 (Green Obsidian)\"" 11 } 12 ok: [greg-rocky86nonlts] =&gt; { 13   "os_version_before.stdout": "PRETTY_NAME=\"Rocky Linux 8.6 (Green Obsidian)\"" 14 } 15 16 TASK [Search to see if LTS is already configured] **** 17 skipping: [greg-rocky86nonlts] 18 ok: [greg-rocky86lts]</pre>					

# Ascender Survey

Templates > SNOW Provision VM > Survey

## Edit Question

**Question \***

**Description**

**Answer type \*** 

Required

**Minimum length**

**Maximum length**

**Answer variable name \*** 

**Default answer**

**Launch | SNOW Provision VM**

1 Other prompts  
2 Survey  
3 Preview

Name of new VM \*  
NewVM1

Choose an OS template to build from \*  
Rocky8.6

Hard drive size in GB \*  
40

Amount of RAM \*  
4096

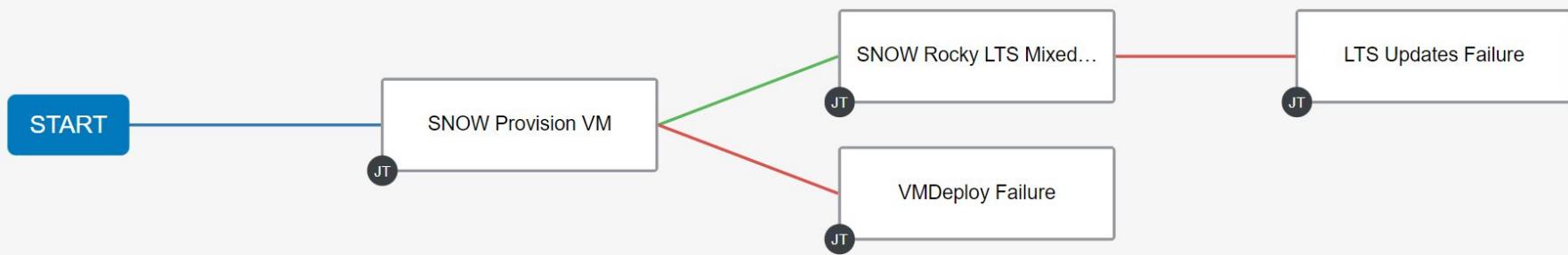
Number of CPUs \*  
4

IP address \*  
10.0.50.24

Network subnet mask \*

**Next** **Back** **Cancel**

# Ascender Workflow



# Ledger/Ledger Pro

Leder is an open source Ascender log collection/processing engine.

- **Ledger**
  - Collects and parses logs
  - Gathers “facts”
  - Tracks changes
  - Easy reporting on gathered info
- **Ledger Pro**
  - Per-host
    - Package monitoring
    - Service status
  - Per-host/Infrastructure
    - Errata
    - CVE
    - Charts
    - Alerts

# Ledger Pro Dashboard



admin ▾

Dashboard

Reports

Hosts

Errata

Changes

Facts

Packages

Services

Admin

Hosts

15

Changes

17

Facts

7195

Packages

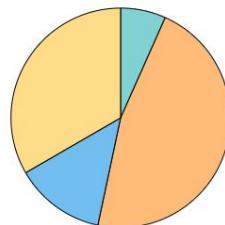
9470

Changes per day



Host Totals by OS

CentOS 7.9    Rocky 8.9    Rocky 9.1    Rocky 9.3



CIQ

# Ledger Pro Facts

The screenshot shows the 'Facts' page in the Ledger Pro interface. On the left is a dark sidebar with navigation links: Dashboard, Reports, Hosts, Errata, Changes, Facts (which is highlighted in green), Packages, Services, and Admin. The main area has a header with 'Hosts', 'Facts', and 'Types' tabs, and search/filter fields for 'jc-books', 'Choose a fact...', 'Choose a type...', and a 'Clear' button. Below this is a table with columns: Host, Fact, Value, and Type. The table lists five facts for the host 'jc-books': ansible\_all\_ipv4\_addresses.0 (Value: 10.0.100.53, Type: gather\_facts), ansible\_apparmor.status (Value: disabled, Type: gather\_facts), ansible\_architecture (Value: x86\_64, Type: gather\_facts), ansible\_bios\_date (Value: 12/12/2018, Type: gather\_facts), and ansible\_bios\_vendor (Value: Phoenix Technologies LTD, Type: gather\_facts). A red box highlights the search/filter section at the top.

Host	Fact	Value	Type
jc-books	ansible_all_ipv4_addresses.0	10.0.100.53	gather_facts
jc-books	ansible_apparmor.status	disabled	gather_facts
jc-books	ansible_architecture	x86_64	gather_facts
jc-books	ansible_bios_date	12/12/2018	gather_facts
jc-books	ansible_bios_vendor	Phoenix Technologies LTD	gather_facts

# Ledger Pro Change Tracking

The screenshot shows the left sidebar of the Ledger Pro application. The menu items are:

- Dashboard
- Reports
- Hosts
- Errata
- Changes** (highlighted with a red border)
- Facts
- Packages
- Services
- Admin

The screenshot shows the main content area of the Ledger Pro application, specifically the Change Tracking table. The table has the following columns:

Date	Host	Playbook	Task	Module
02/19/2024 21:05	jc-demo-rocky8	site.yml	POST   Warning a reboot required but skip option set	ansible.builtin.debug
02/19/2024 21:02	jc-demo-rocky8	site.yml	6.1.1   AUDIT   Audit system file permissions   Add file discrepancy list to system	ansible.builtin.copy

At the top of the table area, there are buttons for "Add a Fil" (dropdown), "Add", and "Filters". Below the table, there are "Show 25 entries" and a "Search: Search in table" input field.

# Ledger Pro Reports

Reports

Hosts

Errata

Changes

Facts

Show 25 entries

Name



[Linux Servers](#)



[Windows Servers](#)

## Linux Servers

[Copy](#) [CSV](#) [Excel](#) [PDF](#) [Print](#)

Show 50 entries

Search:

Showing 1 to 15 of 15 entries

Hostname	IP Address	Distro	Distro Version	Python Version	CPUs	N
AnsibleNinja	10.0.100.57	Rocky	9.3	3.9.18	2	3
Database	10.0.110.35	Rocky	9.3	3.9.18	4	1

Subject

New Report

Enabled



Start

02/20/2024 16:18

Repeat

Daily

Emails

admin

Cancel

Save

Report Name

Linux Servers

Save

Sort Field

Hostname

ASC

Save

Columns

Fact

Display

ansible\_hostname

Hostname



ansible\_default\_ipv4.address

IP Address



ansible\_distribution

Distro



Filters

Fact

Compare

Value

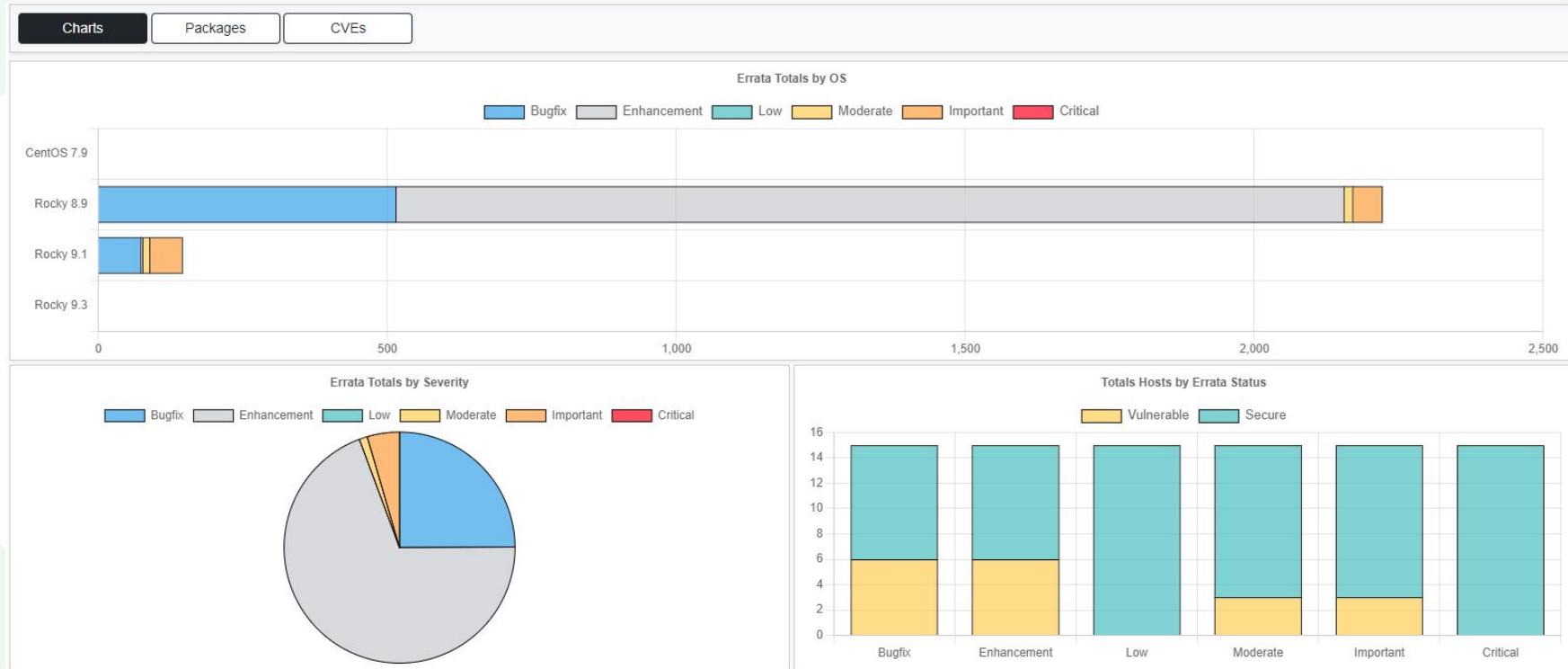
ansible\_system

Equals

Linux

Add Filter

# Ledger Pro Errata



# Ledger Pro Service Status

Hosts	Service	State	Status	Type
jc-books	Choose a service...	Choose a state...	Choose a status...	Clear
Show 25 ▾ entries		Search: <input type="text"/>		
Host	Name	State	Status	Type
jc-books	arp-ethers.service	Inactive	Disabled	Systemd
jc-books	auditd.service	Running	Enabled	Systemd
jc-books	auto-cpufreq.service	Stopped	Not-Found	Systemd
jc-books	autovt@.service	Unknown	Enabled	Systemd
jc-books	blk-availability.service	Inactive	Disabled	Systemd

# Ledger Pro Host Dashboard

ASCENDER  
LEDGER

admin ▾

Dashboard Reports Hosts Errata Changes Facts Packages Services Admin

Show 25 entries Search:

Name	IP Address	OS	Errata	Changes	Facts	Packages
<a href="#">lab-gitea</a>	10.0.100.209	Rocky 9.3		0	474	485
<a href="#">jc-books</a>	10.0.100.53	Rocky 8.9		1	461	863
<a href="#">ansibleninja</a>	10.0.100.57	Rocky 9.3		0	450	490
<a href="#">jc-books-dev-jimmy</a>	10.0.100.58	Rocky 8.9		1	440	994
<a href="#">jc-s40v</a>	10.0.100.59	Rocky 8.9	⭐ 129 🔗 410	1	440	953
<a href="#">jc-books-mark</a>	10.0.100.62	Rocky 8.9	⭐ 129 🔗 410	1	440	808
<a href="#">jc-cacti</a>	10.0.110.10	Rocky 8.9	⭐ 129 🔗 410	1	514	721
<a href="#">jc-demo-rocky9</a>	10.0.110.113	Rocky 9.1	⭐ 37 🔗 2 ⚡ 6 ⚡ 28	0	462	415

# Ledger Pro Host Status

jc-demo-rocky9

Host Info	Packages	Services	Errata
<b>Inventory Hostname (Ascender)</b> jc-demo-rocky9	<b>IPv4 Addresses</b> 10.0.110.113	<b>Hardware</b> <b>Memory (Mb)</b> : 3910 <b>Processors</b> : 2 <b>Processor Cores</b> : 1 <b>Total Cores</b> : 2	
<b>Server FQDN</b> jc-demo-rocky9.ciqlab.dev	<b>IPv6 Addresses</b> fe80::250:56ff:fe87:8c3e	<b>SELinux Status</b> Disabled	
<b>System Vendor</b> VMware, Inc.	<b>Default IPv4</b> <b>Interface</b> : ens192 <b>Address</b> : 10.0.110.113 <b>Netmask</b> : 255.255.255.0 <b>Mac Address</b> : 00:50:56:87:8c:3e <b>Gateway</b> : 10.0.110.1 <b>MTU</b> : 1500 <b>DNS</b> : 10.0.110.1	<b>Python</b> <b>Version</b> : 3.9.14 <b>Path</b> : /usr/bin/python3.9	
<b>Operating System</b> Rocky Linux 9.1		<b>TimeZone</b> CST / CDT (-0600)	
<b>Kernel</b> 5.14.0-162.23.1.el9_1.x86_64			

Disks	Partitions	Mounts
<b>sda</b> <b>Host</b> : Serial Attached SCSI controller: VMware PVSCSI <b>Model</b> : Virtual disk <b>Size</b> : 32.00 GB <b>Sectors</b> : 67108864 (512)	<b>sda1</b> (UUID: C016-F5CD) <b>Size</b> : 600.00 MB <b>Sectors</b> : 1228800 (512)  <b>sda2</b> (UUID: 0704dd31-8a6c-4ba9-acca-85e9b6453d90) <b>Size</b> : 1.00 GB <b>Sectors</b> : 2097152 (512)  <b>sda3</b> (UUID: ) <b>Size</b> : 30.41 GB <b>Sectors</b> : 63778816 (512)	/ (UUID: ed4c915c-8916-4484-a331-22d3a869ee9f) <b>Size</b> : 27.19 GiB  <b>/boot</b> (UUID: 0704dd31-8a6c-4ba9-acca-85e9b6453d90) <b>Size</b> : 1014 MiB  <b>/boot/efi</b> (UUID: C016-F5CD) <b>Size</b> : 598.81 MiB

# Ledger Pro Host Packages Baseline

jc-demo-rocky9

Host Info Package Services Errata

Baseline Diff

Show 25 entries Search:

Name	Errata	Version	Release	Epoch	Arch	Type
bash-completion		2.11	4.el9	1	noarch	rpm
freetype		2.10.4	9.el9		x86_64	rpm
fuse		2.9.9	15.el9		x86_64	rpm
fuse-common		3.10.2	5.el9.0.1		x86_64	rpm
gpg-pubkey		350d275d	6279464b			rpm
graphite2		1.3.14	9.el9		x86_64	rpm

# Ledger Pro Host Services Baseline

jc-demo-rocky9

Host  
Info

Packages

Services

Errata

Baseline Diff

Show 25 ▾ entries

Search:

Name	State	Status	Type
mlocate-updatedb	Stopped	Static	Systemd
systemd-fsck@dev-disk-by-uuid-C016-F5CD	Stopped	Active	Systemd
vgauthd	Running	Enabled	Systemd
vmtoolsd	Running	Enabled	Systemd

# Ledger Pro Host Errata

jc-demo-rocky9

Host Info Packages Services Errata

c-ares-0:1.17.1-5.el9.x86\_64.rpm ⚡ 1 ▲

<b>Advisory :</b> RLSA-2023:3559 <b>Published :</b> 2023-06-13 <b>Severity :</b> ⚠ Important <b>Updated Package :</b> c-ares-0:1.17.1-5.el9_2.1.x86_64.rpm	<b>CVEs:</b> <a href="#">CVE-</a> <a href="#">2023-</a> <a href="#">32067</a>
---	--

**Important: c-ares security update**

The c-ares C library defines asynchronous DNS (Domain Name System) requests and provides name resolving API.

Security Fix(es):

- \* c-ares: 0-byte UDP payload Denial of Service (CVE-2023-32067)

For more details about the security issue(s), including the impact, a CVSS score, acknowledgments, and other related information, refer to the CVE page(s) listed in the References section.

ca-certificates-0:2022.2.54-90.2.el9.noarch.rpm ⚡ 1 ▼

chrony-0:4.2.1.el9.rocky.1.0.x86\_64.rpm ⚡ 1 ▼

dbus-1:1.12.20-7.el9\_1.x86\_64.rpm ▲ 1 ▼



## 1.1 Connect To Lab



Browse To: <https://network.workshop.ciqlab.dev/>

- Enter a unique Name and Email(use info you can remember so you can reconnect)

Lab Name: ascender

---

---

Lab Information

---

Lab Exercises  
Lab Slide Deck

Please enter your Name and Email address.

Name:	<input type="text"/>
Email:	<input type="text"/>
<input type="button" value="Submit"/>	

# Lab Resources

- Lab Exercises - hands on labs
- Ascender - your private Ascender
- VSCode - playbook interface
- GIT - your private git repository

## Lab Information

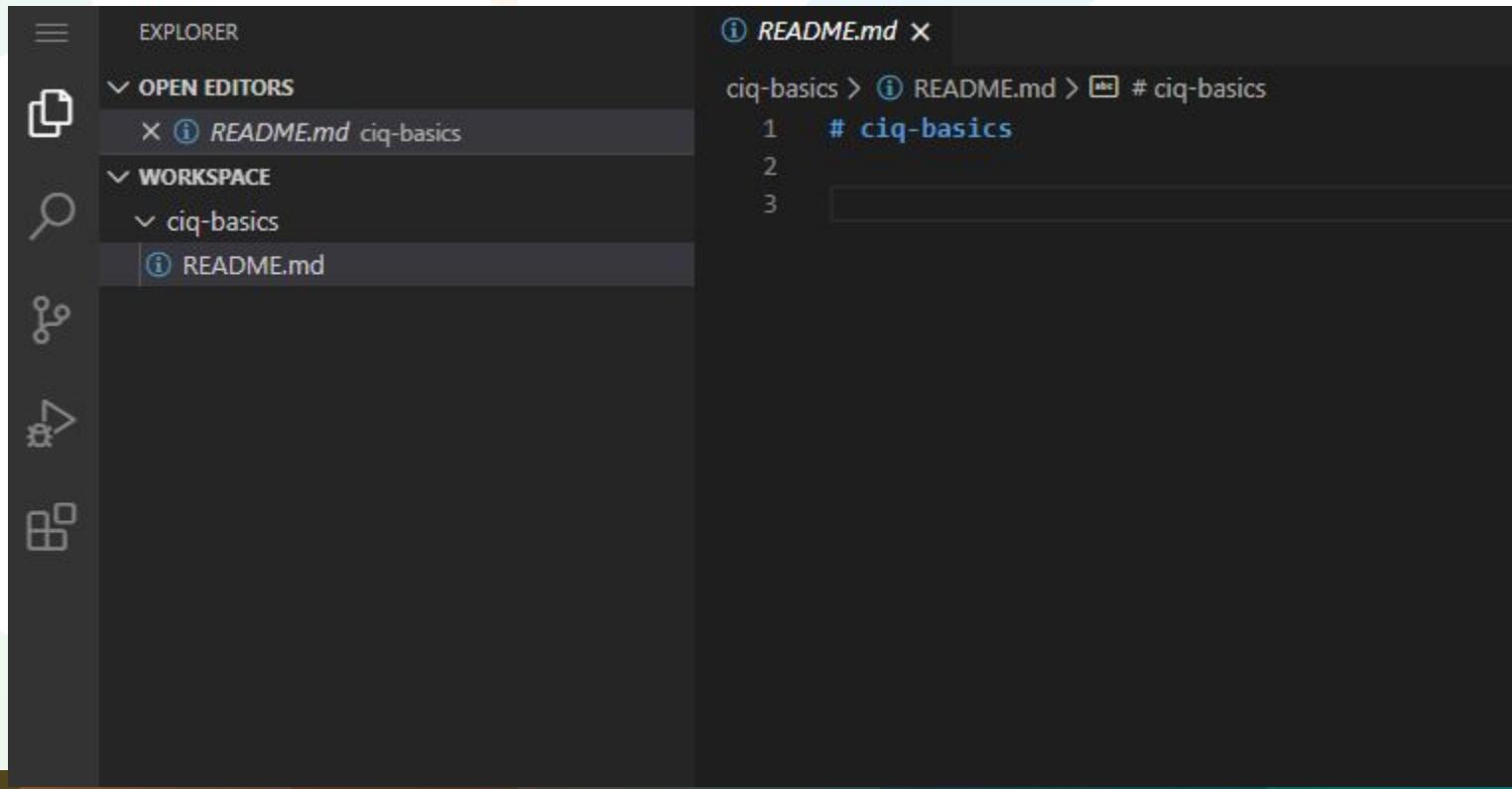
Lab Exercises

Student1

### Lab Nodes

Ascender:	<a href="https://student1.test2.training.ciqlab.dev">https://student1.test2.training.ciqlab.dev</a>
VSCode:	<a href="https://student1-vscode.test2.training.ciqlab.dev">https://student1-vscode.test2.training.ciqlab.dev</a>
GIT:	<a href="https://student1-git.test2.training.ciqlab.dev">https://student1-git.test2.training.ciqlab.dev</a>
username:	student1
password:	RockyLinux!!1

# VSCode Interface



# Gitea Server

All git connections are automatically created in VSCode, so no direct interaction required.

The screenshot shows the Gitea web interface for a repository named 'student1/ciq-basics'. The repository is private, as indicated by the 'Private' button. The main navigation bar includes links for Issues, Pull Requests, Milestones, Explore, and a search icon. On the right side of the header are buttons for Unwatch (1), Star (0), Fork (0), and a gear icon for Settings. Below the header, there's a secondary navigation bar with tabs for Code, Issues, Pull Requests, Packages, Projects, Releases, Wiki, Activity, and Settings. The 'Code' tab is selected. The repository details section shows 'No Description' and 'Manage Topics'. It displays statistics: 1 Commit, 1 Branch, 0 Tags, and 21 KiB. Below these stats are dropdown menus for 'main' branch, 'Go to file', and 'Add File', along with a 'HTTPS' link and a copy icon. A commit history is shown, starting with an 'Initial commit' by user 'student1' at 8a8eaf768, followed by another 'Initial commit' for 'README.md' at the same time. At the bottom, there's a note about 'README.md' and the repository name 'ciq-basics'.



1.1 LAB - 10 mins



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender



## 1.2 Build a simple playbook



# Writing Playbooks

Playbooks are written in YAML

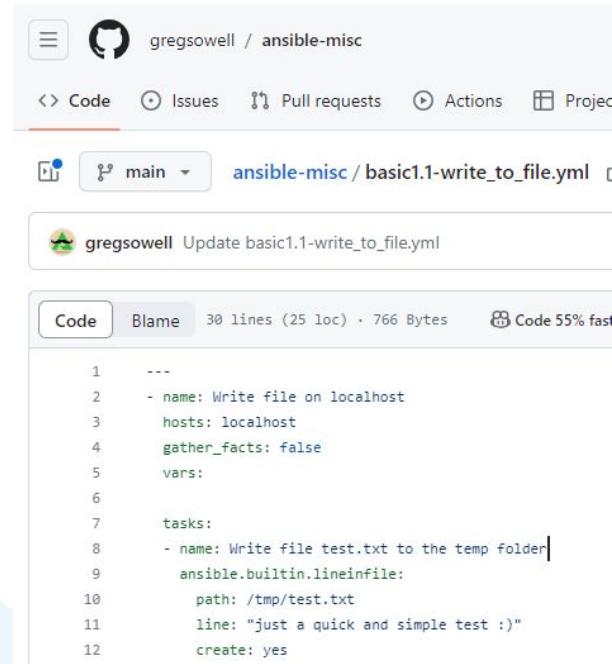
- Human readable
- **Spaces not tabs** - usually 2 spaces

Can be created in multiple ways:

- Simple text editor
  - Notepad++
  - Atom
  - VIM
  - Nano
  - Etc.
- Integrated Development Environment(IDE)
  - Visual Studio Code(VSCode) - Free

# Storing Playbooks

- Playbooks should be stored in a Source Control Management system
  - Git is the most commonly used SCM
  - Ascender utilizes git as a source for its playbooks
  - Vendor agnostic so long as it adheres to git standards



The screenshot shows a GitHub repository interface for a user named gregswell. The repository is named 'ansible-misc'. The 'Code' tab is selected, showing a file named 'basic1.1-write\_to\_file.yml'. The code content is as follows:

```
1 ---  
2 - name: Write file on localhost  
3 hosts: localhost  
4 gather_facts: false  
5 vars:  
6  
7 tasks:  
8 - name: Write file test.txt to the temp folder  
9   ansible.builtin.lineinfile:  
10    path: /tmp/test.txt  
11    line: "just a quick and simple test :)"  
12    create: yes
```

# Our Ansible Playbook

```
---
- name: SNMP updates
  hosts: sw1
  gather_facts: false
  tasks:
    - name: Configure SNMP with the config module
      cisco.ios.ios_config:
        lines:
          - snmp-server community public RO
          - snmp-server community private RW

    - name: Find current SNMP configuration
      cisco.ios.ios_command:
        commands: show run | inc community
        register: snmp_output

    - name: Display current SNMP configuration
      ansible.builtin.debug:
        var: snmp_output.stdout_lines
```

# Ansible Play

- A playbook in Ansible can have multiple plays
  - One play per playbook advantageous for workflows
- Each play must have a new “hosts” section, but it *should* have a name also
- Name portion is often optional, but should be there as part of policy

```
---  
- name: SNMP updates  
  hosts: sw1  
  gather_facts: false
```

# Inventory Vs Hosts

- Inventory
  - File for CLI
  - Database entry in Ascender
  - List of all hosts this job can run against
- Hosts section of playbook
  - Subset of hosts to automate against in inventory specified

```
- name: SNMP updates
  hosts: sw1
  gather_facts: false
```

# Ansible Inventory

An inventory is a list of hosts that Ansible could possibly automate against.

- Can contain **magic** and **custom** variables

```
[web]
webserver1 ansible_host=192.168.1.10 preferred_os=rocky
webserver2 ansible_host=192.168.1.11 preferred_os=rocky
[database]
dbserver1
[loadbalancer]
lbserver1 ansible_user=ansible
```

The diagram illustrates the structure of an Ansible inventory. It shows three host groups: [web], [database], and [loadbalancer]. The [web] group contains two hosts: webserver1 and webserver2. The [database] group contains one host: dbserver1. The [loadbalancer] group contains one host: lbserver1. Two arrows point from the [database] and [loadbalancer] groups towards the [web] group, indicating that dbserver1 and lbserver1 are part of the [web] inventory group.

# Ascender Inventory

Inventories > Rocky LTS Demo

## Hosts

<a href="#">◀ Back to Inventories</a>						
	<a href="#">Details</a>	<a href="#">Access</a>	<a href="#">Groups</a>	<a href="#">Hosts</a>	<a href="#">Sources</a>	<a href="#">Jobs</a>
<input type="checkbox"/>	<input type="checkbox"/> <a href="#">Add</a>	<a href="#">Run Command</a>	<a href="#">Delete</a>		1 - 2 of 2 ▾	
Name	Description	Related Groups				
<input type="checkbox"/> <a href="#">greg-rocky86lts</a>		<a href="#">Its_demo</a>	<input checked="" type="checkbox"/>	On		
<input type="checkbox"/> <a href="#">greg-rocky86nonlts</a>		<a href="#">Its_demo</a>	<input checked="" type="checkbox"/>	On		

# Tasks

- A **task** is an individual item listed under the “tasks” section of a playbook.
- It uses **modules** to perform discrete automation tasks against all “hosts” before proceeding to the next task.

```
tasks:  
  - name: Configure SNMP with the config module  
    cisco.ios.ios_config:  
      lines:  
        - snmp-server community public RO  
        - snmp-server community private RW  
  
  - name: Find current SNMP configuration  
    cisco.ios.ios_command:  
      commands: show run | inc community  
      register: snmp_output  
  
  - name: Display current SNMP configuration  
    ansible.builtin.debug:  
      var: snmp_output.stdout_lines
```

# Network Module Type

- Command
  - Issues commands just as you would from enable mode
  - Not for making changes, only viewing information
  - Show run, show ver, show ip bgp nei
- Config
  - Makes configuration changes via standard CLI commands
  - Only idempotent if you use full command syntax
- Resource
  - Application specific configurations
  - Fully idempotent
  - ios\_acls, ios\_hostname, ios\_snmp\_server



1.2 Lab - 20 mins





## 1.3 Ascender Job Template Creation



# Credentials

The screenshot shows the Ascender Automation web application interface. The top navigation bar includes the Ascender Automation logo, a bell icon, a help icon, and a user account dropdown for 'student1'. The left sidebar has two main sections: 'Views' (Dashboard, Jobs, Schedules, Activity Stream, Workflow Approvals) and 'Resources' (Templates, Credentials, Projects, Inventories, Hosts). The 'Credentials' option under 'Resources' is highlighted with a green background. The main content area is titled 'Credentials' and displays a list of three entries:

Name	Type	Actions
<a href="#">Ansible Galaxy</a>	Ansible Galaxy/Automation Hub API Token	<a href="#"></a> <a href="#"></a>
<a href="#">basics-credential</a>	Machine	<a href="#"></a> <a href="#"></a>
<a href="#">scm-basics-course</a>	Source Control	<a href="#"></a> <a href="#"></a>

Pagination at the bottom indicates there are 1 - 3 of 3 items, 1 of 1 page.

# Projects

The screenshot shows the Ascender Automation software interface. At the top, there is a navigation bar with the Ascender Automation logo, a bell icon, a help icon, and a user account for "student1". On the left side, there is a sidebar with "Views" and "Resources" sections. Under "Views", the options are "Dashboard", "Jobs", "Schedules", "Activity Stream", and "Workflow Approvals". Under "Resources", the options are "Templates", "Credentials", "Projects" (which is highlighted in green), and "Inventories". The main content area is titled "Projects" and displays a table of projects. The table has columns: Name, Status, Type, Revision, and Actions. One project is listed: "basics-course" (Status: Successful, Type: Git, Revision: b295ca2). There are buttons for "Add" and "Delete" at the top of the table, and pagination controls at the bottom.

Name	Status	Type	Revision	Actions
basics-course	Successful	Git	b295ca2	

# Inventories



student1

- Views
- Dashboard
  - Jobs
  - Schedules
  - Activity Stream
  - Workflow Approvals

- Resources
- Templates
  - Credentials
  - Projects
  - Inventories

## Inventories

<input type="checkbox"/> Name	Sync Status	Type	Organization	Actions
<input type="checkbox"/> basics-inventory	Disabled	Inventory	Default	

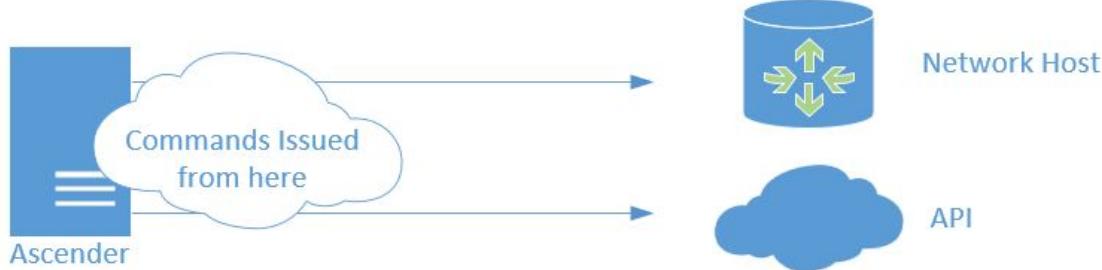
1 - 1 of 1 items ▾    <<    <    1 of 1 page    >    >>

CIQ

# Local Vs Remote Execution

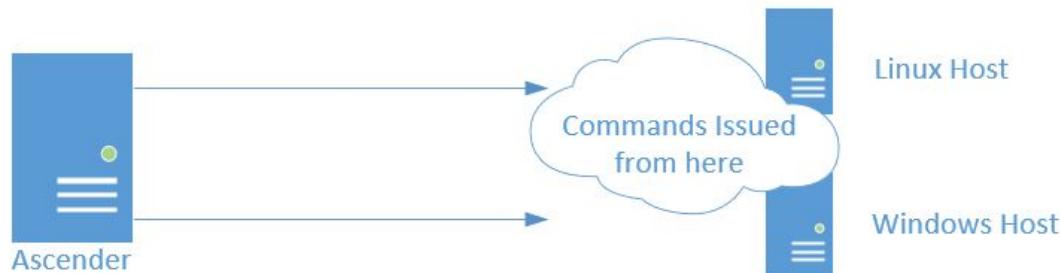
## Local Execution

Ascender issues commands directly from itself to the remote host



## Remote Execution

Code/files are copied to remote host and executed there



# Connection Plugins - Network

Use `ansible_connection` variable to modify for network devices:

- `network_cli` - Standard SSH CLI interface(most commonly used)
  - Arista EOS, Cisco IOS, Cisco NXOS
- `netconf` - XML over netconf
  - Juniper Junos
- `httpapi` - Vendor specific API
  - Cisco NX-API or Arista eAPI

# Network OS

- Tells Ansible what language to speak to the device(english, french, spanish)
- Variable is `ansible_network_os`
  - Cisco IOS = `ansible_network_os: ios`
  - Cisco NXOS = `ansible_network_os: nxos`
  - Arista EOS = `ansible_network_os: eos`

# Host Variables

Name rtr1

Created 6/4/2024, 5:07:55 PM by admin

Last Modified 6/4

Variables

YAML

JSON

```
1 ansible_host: 172.20.20.2
2 ansible_connection: network_cli
3 ansible_network_os: ios
4 ansible_ssh_common_args: '-o KexAlgorithms=+diffie-hellman-group1-sha1 -o HostKeyAlgorithms=+ssh-rsa -o Ciphers=+aes256-cbc'
```

# Templates

ASCENDER AUTOMATION

student1

Views

- Dashboard
- Jobs
- Schedules
- Activity Stream
- Workflow Approvals

Resources

- Templates
- Credentials
- Projects

Templates

Name	Activity	Last Ran	Type	Actions
basics-conditionals	✓ ✓ ✓	2/26/2024, 3:54:28 PM	Job Template	
basics-handlers	✓ ! !	2/26/2024, 4:26:18 PM	Job Template	

# Project URL

Copy and paste YOUR PERSONAL git url here, and then add on “/YourStudentID/ciq-basics.git”, but replace YourStudentID with your actual student ID.

So, student12 would be “/student12/ciq-basics.git”

Source Control URL	Copy and paste the URL of your git server and add “/YOURSTUDENTID/ciq-basics.git” to it.  For student1, this would look like <a href="https://student1-git.test2.training.ciqlab.dev/student1/ciq-basics.git">https://student1-git.test2.training.ciqlab.dev/student1/ciq-basics.git</a>
--------------------	---



1.3 Lab - 20 mins





## 1.4 Execute a playbook



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender

# Launch Ansible Playbook

The screenshot shows the Ascender Automation web application interface. On the left, there is a navigation sidebar with the following menu items:

- Views: Dashboard, Jobs, Schedules, Activity Stream, Workflow Approvals.
- Resources: Templates (selected), Credentials, Projects, Inventories, Hosts.
- Access: Organizations, Users, Teams.
- Administration: Credential Types, Notifications.

The main content area has the following details:

- Template Path: Templates > basics-conditionals
- Section: Details
- Buttons: Back to Templates, Details, Access, Edit, Launch (highlighted with a red box), Delete.
- Form Fields:
  - Name: basics-cond
  - Organization: Default
  - Execution Environment: Ascender-Environ
  - Verbosity: 0 (Normal)
  - Job Slicing: 1
  - Credentials: SSH: basics-cond
  - Variables: YAML (highlighted with a red box), JSON
- Table Row (Bottom):
  - Index: 1
  - Value: ---
  - Actions: Edit, Launch (highlighted with a red box), Delete.

To the right, there is a large preview window titled "ASCENDER AUTOMATION" showing the "Templates" section. It includes a search bar, an "Add" button, and a table with the following data:

Name	Activity	Last Ran	Type	Actions
basics-conditionals	✓ ✓ ✓	2/26/2024, 3:54:28 PM	Job Template	

# Idempotence

- Run playbooks multiple times and only make a change if required

Jobs > 15 - basics-snmp-config

## Output

A screenshot of the Ansible Playbook output interface. The top navigation bar shows 'Back to Jobs', 'Details', and 'Output' (which is highlighted with a red box). Below the navigation, the job details are shown: 'basics-snmp-config' (Successful), Plays 1, Tasks 3, Hosts 1, Elapsed 00:00:11. A large yellow progress bar is present. The main area is titled 'Stdout' with a dropdown menu and a search icon. The log output is displayed in a table:

0	SSH password:
1	
2	PLAY [SNMP updates] **** 6:56:43

Jobs > 28 - basics-snmp-config

## Output

A screenshot of the Ansible Playbook output interface. The top navigation bar shows 'Back to Jobs', 'Details', and 'Output' (which is highlighted with a blue underline). Below the navigation, the job details are shown: 'basics-snmp-config' (Successful), Plays 1, Tasks 3. A large green progress bar is present. The main area is titled 'Stdout' with a dropdown menu and a search icon. The log output is displayed in a table:

0	SSH password:
1	
2	PLAY [SNMP updates] ****
3	
4	TASK [Configure SNMP with the config module] ***
5	ok: [rtr1]
6	

## Idempotence - Config Modules

- Must use full command to maintain idempotence
  - If you use abbreviations it will show “changed” every time

# Output

<code>ok</code>	Everything executed fine, but no changes were made.
<code>changed</code>	Everything ran fine, but it appears that a change was made.
<code>failed</code>	The task failed for a host. This will cause the playbook to show a failed status.
<code>skipping</code>	This task had a conditional set, and the host didn't match, so it skipped for this specific host.



1.4 Lab - 10 mins



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender



## 1.5 Resource Modules



# Resource Modules

- Application specific configuration with baked in idempotency
- EX: ios\_acls, ios\_banner, ios\_lacp
- Several interesting states
  - merged - (general default) add this to the existing configuration
  - replaced - remove whatever config is there and place this instead
  - overridden - modify/add the configs defined, deleted all others
  - deleted - deletes specified config
  - gathered - fetch the running configuration from device and transform it into structured data in the format as per the resource module
  - rendered - transform the configuration in config option to platform specific CLI commands
  - parsed - takes cli formatted commands and parses them into the module's data model

# Power Of Parsed

- Take standard CLI commands, parse them, apply them

```
***snmp.txt file  
snmp-server community public RO  
snmp-server community private RW
```

```
- name: Parse config  
cisco.ios.ios_snmp_server:  
  running_config: "{{ lookup('file', 'files/snmp.txt') }}"  
  state: parsed  
  register: parsed
```

```
- name: Replace existing SNMP  
configuration with the ios_snmp_server  
module  
cisco.ios.ios_snmp_server:  
  config: "{{ parsed.parsed }}"  
  state: replaced
```



1.5 Lab - 20 mins





## 1.6 Variables And Loops



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender

# Why Use Variable

Variables should be used frequently

- Allow for maximum flexibility in playbooks
  - Prevent regular modification
  - Facilitates sharing playbooks
  - Provides dynamic reactions

# Variable Basics

- Can only contain numbers, letters, and underscores
- Can not be numbers only
- Can not begin with numbers
- Case sensitive

# Creating/Referencing Variables

- Variables can be created in a “vars” section like below
  - variable\_name: contents of variable
    - The same format is used if variables are stored in an external file
- Variables are, generally, referenced via the following format
  - “{{ name }}”
    - hd\_size: “{{ hd\_gb\_size }}”
    - error\_msg: “The error message is: {{ err\_returned }}”

```
vars:  
  hd_gb_size: 10  
  
tasks:  
- name: Display disk size  
  ansible.builtin.debug:  
    msg: "{{ hd_gb_size }}"
```

# Register A Variable

- The output of any task can be saved to a variable via the “**register**” option
  - register: name\_of\_variable
- Registering a variable creates a new or overwrites an existing variable
- The contents of the variable will be uniquely saved for each host this runs against

```
- name: Grab contents of file
  ansible.builtin.shell: cat /tmp/test.txt
  register: file_contents

- name: Display file_contents
  ansible.builtin.debug:
    msg: "{{ file_contents.stdout }}"
```

# Lists

- Lists are variables that contain a list of items
- Works similarly to an array
- Can be iterated over easily in a loop
- Individual items referenced with list\_name[#]

```
list1:  
  - thing1  
  - thing2  
  - thing3  
  
  - name: pulling items from list 1  
ansible.builtin.debug:  
  msg: "{{ list1[1] }}"
```

# Dictionary

- Stores multiple pieces of information under a single variable name
- Not iterable by default
- Referenced in one of two ways:
  - `dict3.1.name`
    - Simplest, but can conflict with python keywords
  - `dict3['1']['name']`
    - More characters, but conflict free

```
dict3:  
1:  
    name: New Greg  
    color: Bright  
2:  
    name: Andy  
    color: Green  
3:  
    name: Andrew  
    color: Baby blue
```

# Advanced Filtering/Looping With Lists/Dictionaries

- Greg Sowell's blog post: <https://gregsowell.com/?p=7380>

## Vars Section

- Vars section can exist for a play or individual task
- Good practice to keep all variables in vars section(even if they are to be overwritten)

```
- name: test playbook
hosts: web
vars:
  new_var1: contents of my variable
```

# Variable Precedence

## Precedence from low to high

1. command line values (for example, -u my\_user, these are not variables)
2. role defaults (defined in role/defaults/main.yml)
3. inventory file or script group vars
4. inventory group\_vars/all
5. playbook group\_vars/all
6. inventory group\_vars/\*
7. playbook group\_vars/\*
8. inventory file or script host vars
9. inventory host\_vars/\*
10. playbook host\_vars/\*
11. host facts / cached set\_facts
12. play vars
13. play vars\_prompt
14. play vars\_files
15. role vars (defined in role/vars/main.yml)
16. block vars (only for tasks in block)
17. task vars (only for the task)
18. include\_vars
19. set\_facts / registered vars
20. role (and include\_role) params
21. include params
22. extra vars (for example, -e "user=my\_user") (always win precedence)

## Extra Vars

- Extra Vars have the highest level of precedence, thus overwrite any existing vars with same name
- Available via CLI or Ascender

### CLI

```
ansible-playbook this.yml --extra-vars '{"name":"bob","fave_col":"red"}'
```

# Extra Vars Ascender

- Kept with job template
- Can be added via:
  - Hard coded
  - Prompt on launch
  - Collected from user via survey
  - Accepted via API call

Templates > CAC Nexus Config

## Details

◀ Back to Templates      Details      Access

Name: CAC Nexus Config

Inventory: Network Lab

Variables (YAML)

```
1 ---  
2 backup_with_tags: true
```

# Loops

- The loop option allows for iterating over a list
- List can be static or dynamic
- “Loop” keyword is new, “with\_items” is the old way

# Inefficient Example

```
---
- name: Add Users
  hosts: localhost
  gather_facts: false

  tasks:
    - name: Add a user1
      ansible.builtin.user:
        name: user1
        state: present

    - name: Add a user2
      ansible.builtin.user:
        name: user2
        state: present

    - name: Add a user3
      ansible.builtin.user3:
        name: user3
        state: present
```

# Loop Example Static List

- “item” is the default return variable name of a loop
- Each iteration of the loop replaces the “item” variable with a value

```
---
- name: Add Users
hosts: localhost
gather_facts: false

tasks:
- name: Add users
ansible.builtin.user:
  name: "{{ item }}"
  state: present
loop:
- user1
- user2
- user3
```

# Loop Example Variable

- Loop is most often supplied via variable
- Variable often defined:
  - Static in playbook
  - Variable file
  - Settings in role
  - Extra vars
  - Dynamically via another task

```
---
- name: Add Users
hosts: localhost
gather_facts: false
vars:
  user_list:
    - user1
    - user2
    - user3

tasks:
- name: Add users
  ansible.builtin.user:
    name: "{{ item }}"
    state: present
  loop: "{{ user_list }}"
```



1.6 Lab - 20 mins





## 1.7 Facts and Conditionals



# Network Facts

- ios\_facts, eos\_facts, nxos\_facts, etc.
- Designed for the specific network OS queried
- Configurable amount of variablized data returned from network host
  - More than just “show version”
- Still requires “gather\_facts: false” at top of playbook

```
- name: Gather facts from the router
  cisco.ios.ios_facts:
    gather_subset: min
    gather_network_resources: snmp_server
  register: facts
```

# What do network facts look like?

```
root@workshop-network:~ - □ ×
Cisco IOS XE Software, Version 16.06.01
Cisco IOS Software [Everest], Virtual XE Software (X86_64_LINUX_IOSD-UNIVERSALK9-M), Version 16.6.1, RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2017 by Cisco Systems, Inc.
Compiled Sat 22-Jul-17 05:51 by mcpre

Cisco IOS-XE software, Copyright (c) 2005-2017 by cisco Systems, Inc.
All rights reserved. Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0. The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY. You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0. For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.

ROM: IOS-XE ROMMON

rtr1 uptime is 2 weeks, 6 days, 18 hours, 34 minutes
Uptime for this control processor is 2 weeks, 6 days, 18 hours, 37 minutes
```

Show Version

```
ok: [rtr1] => {
  "facts": {
    "ansible_facts": {
      "ansible_net_api": "cliconf",
      "ansible_net_gather_network_resources": [
        "snmp_server"
      ],
      "ansible_net_gather_subset": [
        "default"
      ],
      "ansible_net_hostname": "rtr1",
      "ansible_net_image": "bootflash:packages.conf",
      "ansible_net_iostype": "IOS-XE",
      "ansible_net_model": "CSR1000V",
      "ansible_net_operatingmode": "controller",
      "ansible_net_python_version": "3.8.16",
      "ansible_net_serialnum": "9DM0ADJEHVE",
      "ansible_net_system": "ios",
      "ansible_net_version": "16.06.01",
```

ios\_facts

# Conditionals

- Conditionals all for single or multiple true/false conditions to be set on tasks
- It's generally when:
  - Value of one variable is compared to that of another
    - `var1 == var2`
  - Value of a variable is compared to some static value
    - `var1 == 5`

# Conditional Example 1

- Simple mathematical comparison
- Compares numbers or strings
- **Notice no quotes or curly braces for variables in the conditional**
  - When Ansible expects Jinja2 no quotes or braces are needed for variables

```
vars:  
    tacos: 5  
  
tasks:  
    - name: Print taco time  
      ansible.builtin.debug:  
        msg: "we have plenty of tacos"  
      when: tacos >= 5
```

## Conditional Example 2

- Compound comparison

```
vars:  
  tacos: 5  
  day: Tuesday  
  
tasks:  
  - name: Print taco time  
    ansible.builtin.debug:  
      msg: "we have plenty of tacos"  
    when: tacos >= 5 and day == "Tuesday"
```



1.7 Lab - 15 mins





## 2.1 Surveys

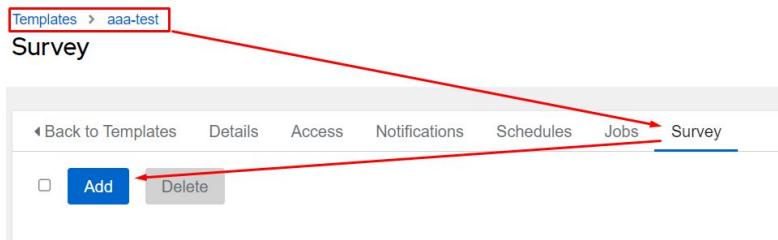


# Survey Basics

- Prompt the user for additional info at automation run
- Simple to build question/answer forms
  - Question types
    - Text
    - Text area
    - Password
    - Multiple choice(single or multi select)
    - Integer
    - Float
- Answers are passed to playbook as extra\_vars(highest level of preference)

# Creating Surveys

- Enter a template, click Survey tab, and Add



## Add Question

The 'Add Question' form is displayed. It includes the following fields:

- Question \***: An empty text input field.
- Description**: An empty text input field.
- Answer variable name \***: An empty text input field.
- Answer type \***: A dropdown menu set to 'Text'.
- Required**: A checked checkbox.
- Minimum length**: A text input field containing '0'.
- Maximum length**: A text input field containing '1024'.
- Default answer**: An empty text input field.

# Surveys Question Order

- Click Edit Order and drag the questions around

	Name	Type	Default	Actions
<input type="checkbox"/>	What VM action should be performed *	multiplechoice	provision	
<input type="checkbox"/>	Name of new VM *	text	test-rocky9	
<input type="checkbox"/>	Choose an OS template to build from *	multiplechoice	Rocky9	
<input type="checkbox"/>	Hard drive size in GB *	multiplechoice	40	
<input type="checkbox"/>	Amount of RAM *	multiplechoice	4096	

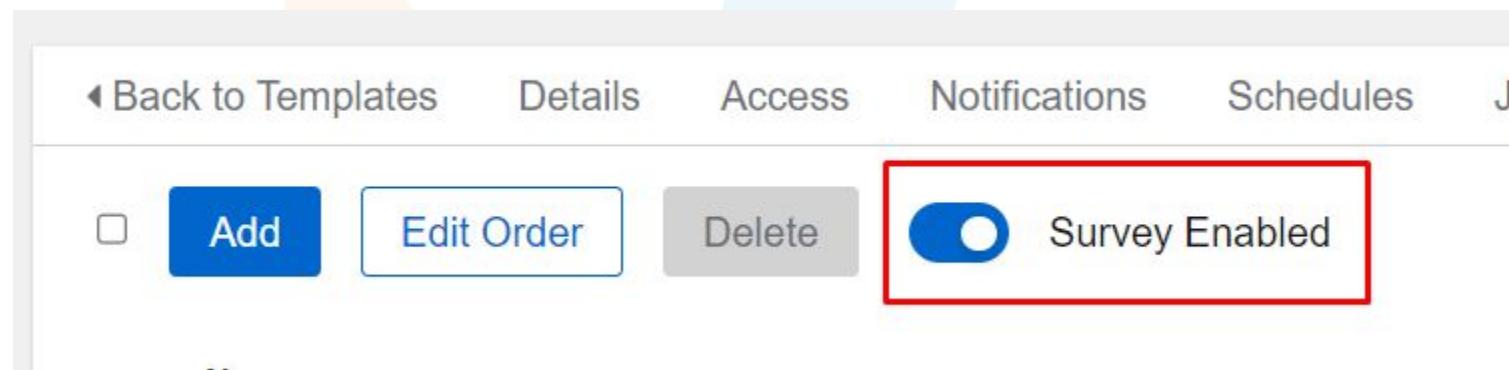
Survey Question Order

To reorder the survey questions drag and drop them in the desired location.

Order	Name	Default Answer(s)
#	What VM action should be performed	provision
#	Name of new VM	test-rocky9
#	Choose an OS template to build from	Rocky9
#	Hard drive size in GB	40
#	Amount of RAM	4096

## Enable Surveys

- By default surveys are disabled. If you want them to be visible, click the slider button until it says “Survey Enabled”



# Launch Survey

- Launch job template and survey screen pops up
- Next screen shows variables

Launch | VMWare VM Control

1 Survey      2 Preview

What VM action should be performed \*

provision

Name of new VM \*

test-rocky9

Choose an OS template to build from \*

Rocky9

Hard drive size in GB \*

40

Amount of RAM \*

4096

Number of CPUs \*

4

Next Back Cancel

Launch | VMWare VM Control

1 Survey      2 Preview

Forks 0      V

Job 1      Slicing

Credentials      SSH: Lab Server...      Cloud: VM

Created 2/13/2024, 3:17:11 PM by admin

Prompted Values

Variables      YAML      JSON

```
1 vm_action: provision
2 vm_name: test-rocky9
3 vm_template: Rocky9
4 vm_disksize: '40'
```



2.1 Lab - 15 mins



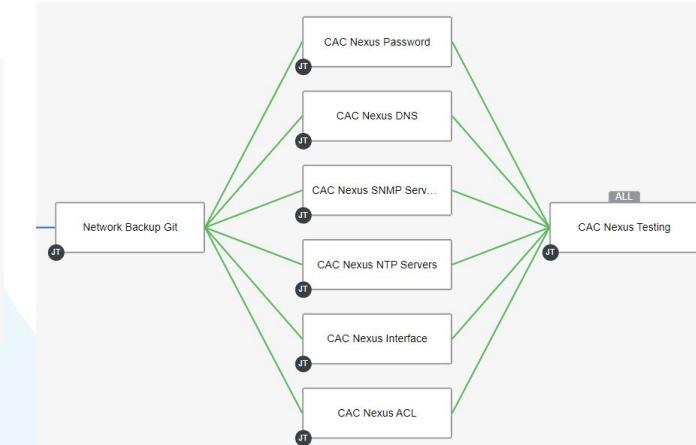
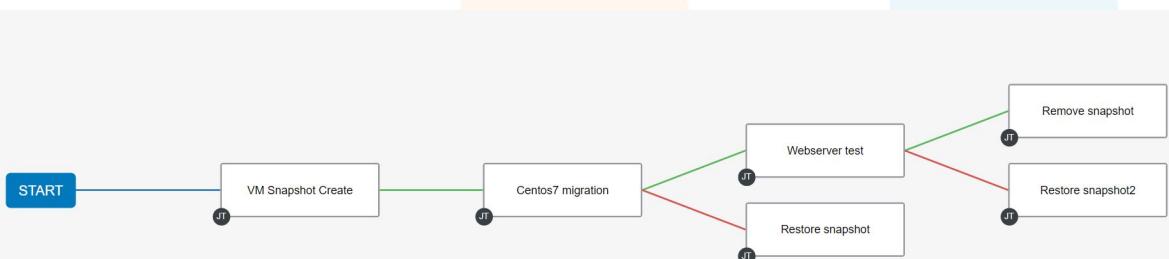


## 2.2 Workflows



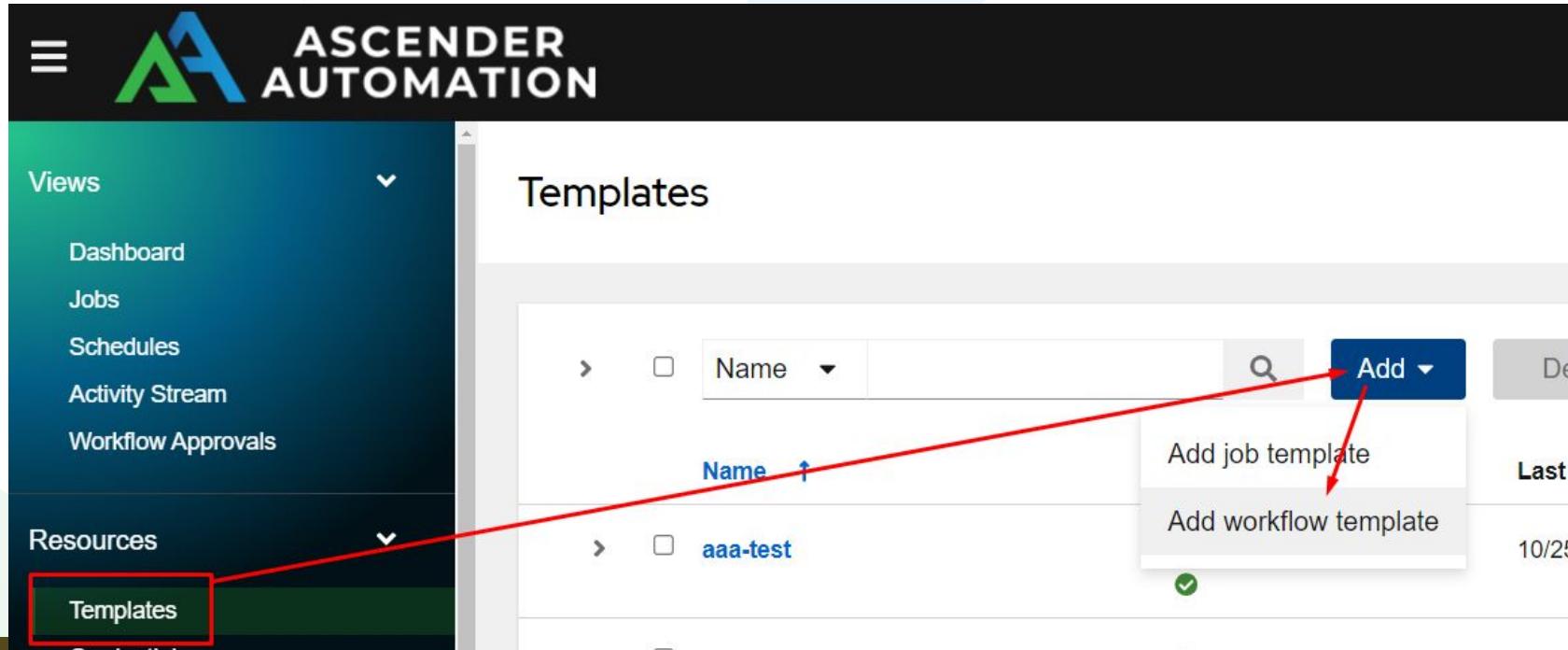
# Workflow Basics

- Visually chain together job templates
- Branching paths for: on success, on failure, always
- Parallel processing
- Merging job template paths
- Cross silo collaboration - share job templates



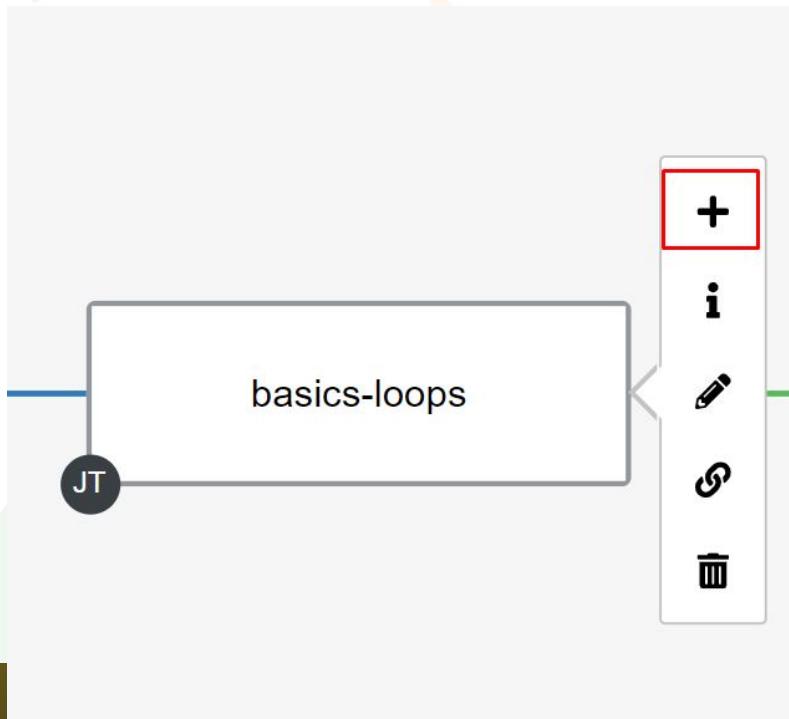
# Creating Workflows

- Enter a template, Add, and choose Add workflow template



# Adding Items

- Select when to run a job



### Add Node

1 Run type  
2 Node type

Run  
Specify the conditions under which this node should be executed

**On Success**  
Execute when the parent node results in a successful state.

**On Failure**  
Execute when the parent node results in a failure state.

**Always**  
Execute regardless of the parent node's final state.

### Add Node

Node Type	Job Template
Name	<input type="text"/>
Name ↑	<input type="radio"/> aaa-test <input type="radio"/> Ascender CAC Migration Export <input type="radio"/> Ascender CAC Migration Import <input type="radio"/> basics-loops <input type="radio"/> basics-write-file-vars-survey
Convergence *	<input type="text"/> Any



2.2 Lab - 20 mins





[info@ciq.com](mailto:info@ciq.com) - (800) 220-5243

**Good luck, keep learning!**