



# Ansible/Ascender Basics



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender

# 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



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender

## 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

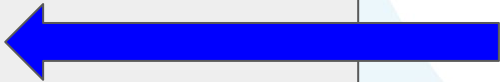
# Ansible Playbooks

```
---  
- 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
```




# 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
```

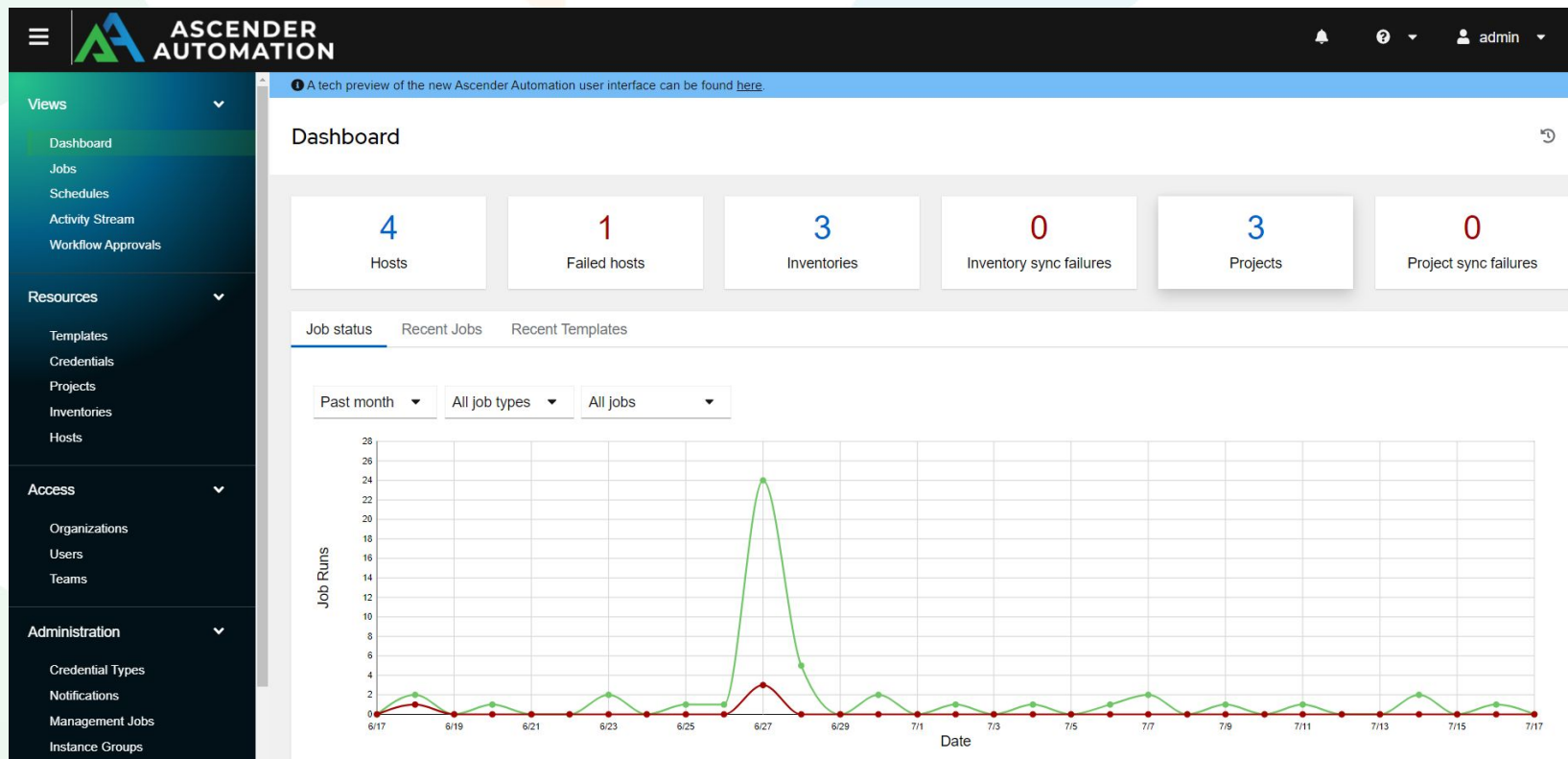


# Ansible Collections

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



# 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 interface, illustrating its Role-Based Access Control (RBAC) capabilities.

**Left Screenshot: Users Management**

- Header:** ASCENDER AUTOMATION
- Left Sidebar:** Contains navigation links for Templates, Credentials, Projects, Inventories, Hosts, and a dropdown menu for Access (which includes Organizations, Users, and Teams).
- Main Content Area:** Titled "Users", it features a search bar with a dropdown menu set to "Email", a search icon, and buttons for "Add" and "Delete". Below this is a table with columns for Username, First Name, Last Name, and Role. One user is listed: "admin" with the role "System Administrator".

**Right Screenshot: Settings**

- Header:** ASCENDER AUTOMATION
- Left Sidebar:** Contains navigation links for Access (Organizations, Users, Teams) and Administration (Credential Types, Notifications, Management Jobs, Instance Groups, Instances, Applications, Execution Environments, Topology View), with a "Settings" link at the bottom.
- Main Content Area:** Titled "Settings", it includes a section for "Authentication" with the description "Enable simplified login for your Ascender Automation applications". Below this are several settings links: Azure AD settings, GitHub settings, Google OAuth 2 settings, LDAP settings, RADIUS settings, SAML settings, TACACS+ settings, and Generic OIDC settings.

# Ascender Inventories

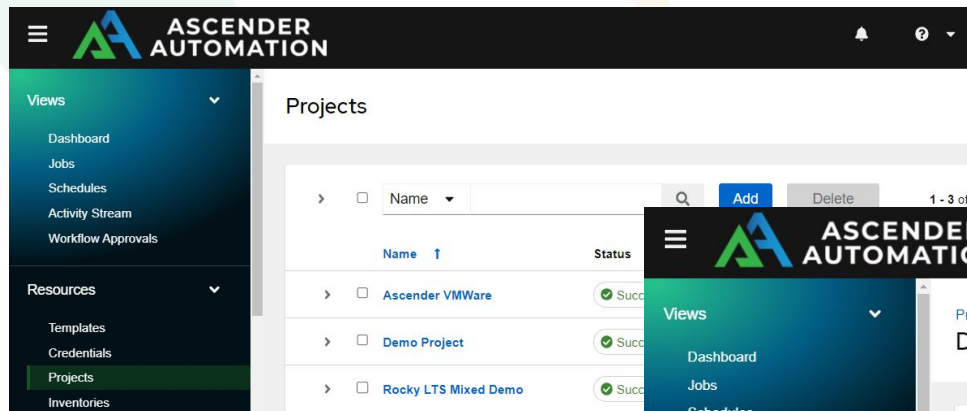
This screenshot shows the 'Inventories' page in the Ascender Automation interface. The left sidebar contains a menu with 'Views' (Dashboard, Jobs, Schedules, Activity Stream, Workflow Approvals) and 'Resources' (Templates, Credentials, Projects, Inventories, Hosts). The 'Inventories' section is active. The main content area has a search bar with a dropdown menu set to 'Name', a search icon, and buttons for 'Add' and 'Delete'. Below the search bar is a table listing three inventory items:

	Name ↑
<input type="checkbox"/>	Demo Inventory
<input type="checkbox"/>	Rocky LTS Demo
<input type="checkbox"/>	SNOW Provision VM

This screenshot shows the 'Hosts' page within the 'Rocky LTS Demo' inventory. The left sidebar is identical to the first screenshot, with 'Hosts' selected under the 'Resources' section. The main content area has a breadcrumb trail 'Inventories > Rocky LTS Demo' and a title 'Hosts'. Below the title is a navigation bar with tabs: 'Back to Inventories', 'Details', 'Access', 'Groups', 'Hosts' (active), 'Sources', and 'Jobs'. The 'Hosts' tab contains a search bar with a dropdown menu set to 'Name', a search icon, and buttons for 'Add' and 'Run Command'. Below the search bar is a table listing two hosts:

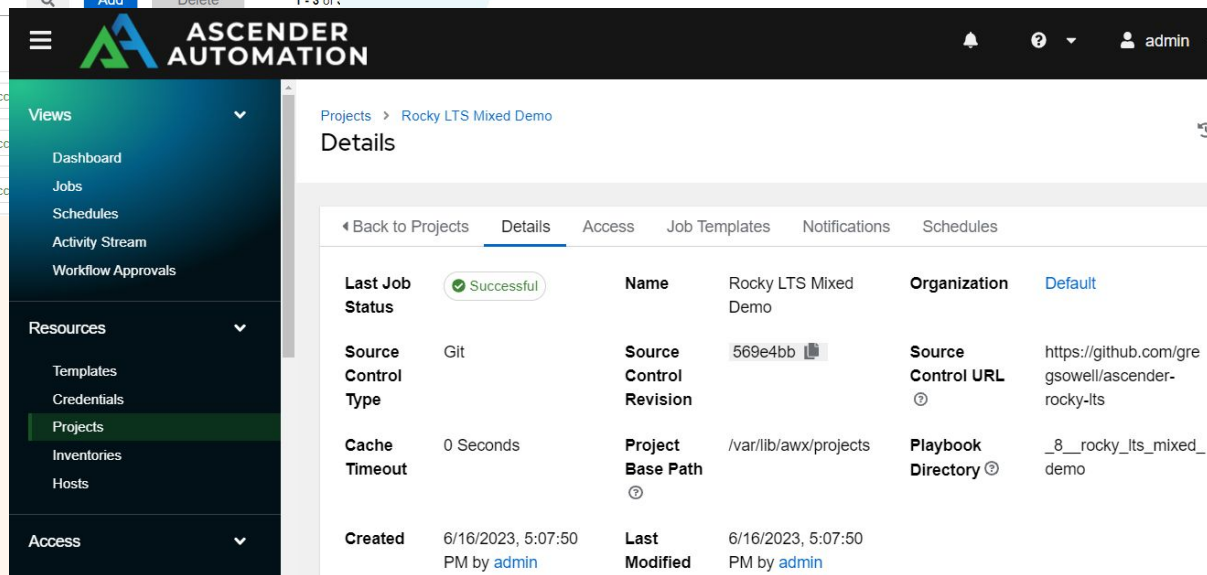
	Name ↑	Description ↓	Related Groups
<input type="checkbox"/>	greg-rocky86its		Its_demo
<input type="checkbox"/>	greg-rocky86nonits		Its_demo

# Ascender Projects



This screenshot shows the 'Projects' page in the Ascender Automation interface. The left sidebar contains a 'Views' menu with options: Dashboard, Jobs, Schedules, Activity Stream, and Workflow Approvals. Below this is a 'Resources' menu with options: Templates, Credentials, Projects (highlighted), and Inventories. The main content area is titled 'Projects' and displays a table with three projects: 'Ascender VMWare', 'Demo Project', and 'Rocky LTS Mixed Demo'. Each project has a checkbox, a status indicator (green checkmark), and a 'Success' label. The table has columns for 'Name' and 'Status'.

Name	Status
Ascender VMWare	Success
Demo Project	Success
Rocky LTS Mixed Demo	Success

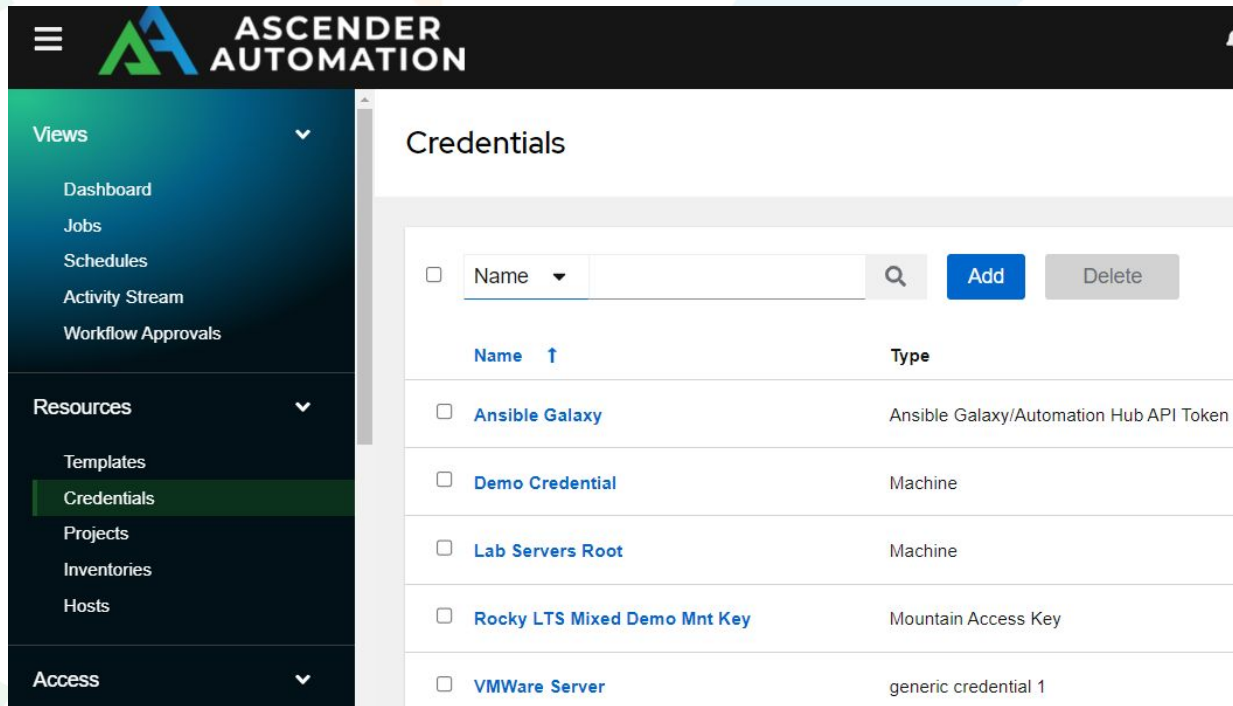


This screenshot shows the 'Details' page for the 'Rocky LTS Mixed Demo' project in the Ascender Automation interface. The left sidebar is identical to the previous screenshot, with 'Projects' highlighted in the 'Resources' menu. The main content area is titled 'Details' and shows a table with project information. The table has columns for 'Last Job Status', 'Source Control Type', 'Cache Timeout', 'Created', 'Name', 'Source Control Revision', 'Project Base Path', 'Last Modified', 'Organization', 'Source Control URL', and 'Playbook Directory'. The 'Last Job Status' is 'Successful'. The 'Source Control Type' is 'Git'. The 'Cache Timeout' is '0 Seconds'. The 'Created' date is '6/16/2023, 5:07:50 PM by admin'. The 'Name' is 'Rocky LTS Mixed Demo'. The 'Source Control Revision' is '569e4bb'. The 'Project Base Path' is '/var/lib/awx/projects'. The 'Last Modified' date is '6/16/2023, 5:07:50 PM by admin'. The 'Organization' is 'Default'. The 'Source Control URL' is 'https://github.com/gsgowell/ascender-rocky-lts'. The 'Playbook Directory' is '\_8\_\_rocky\_lts\_mixed\_demo'.

Last Job Status	Source Control Type	Cache Timeout	Created	Name	Source Control Revision	Project Base Path	Last Modified	Organization	Source Control URL	Playbook Directory
Successful	Git	0 Seconds	6/16/2023, 5:07:50 PM by admin	Rocky LTS Mixed Demo	569e4bb	/var/lib/awx/projects	6/16/2023, 5:07:50 PM by admin	Default	https://github.com/gsgowell/ascender-rocky-lts	_8__rocky_lts_mixed_demo



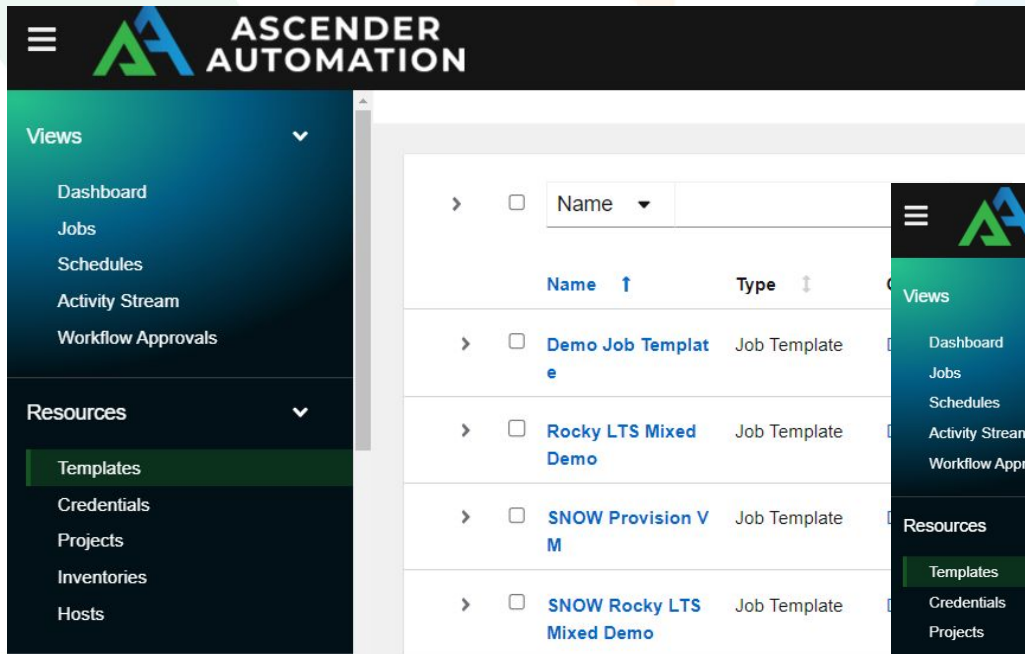
# Ascender Credentials



The screenshot displays the Ascender Automation web interface. The top navigation bar is black with the Ascender Automation logo on the left and a notification bell on the right. A left sidebar contains a menu with sections: Views (Dashboard, Jobs, Schedules, Activity Stream, Workflow Approvals), Resources (Templates, Credentials, Projects, Inventories, Hosts), and Access. The 'Credentials' page is active, showing a table of credentials. At the top of the table is a search bar with a dropdown menu set to 'Name', a search icon, and buttons for 'Add' and 'Delete'. The table has two columns: 'Name' and 'Type'. It lists five credentials: 'Ansible Galaxy' (Ansible Galaxy/Automation Hub API Token), 'Demo Credential' (Machine), 'Lab Servers Root' (Machine), 'Rocky LTS Mixed Demo Mnt Key' (Mountain Access Key), and 'VMWare Server' (generic credential 1). Each row has a checkbox on the left.

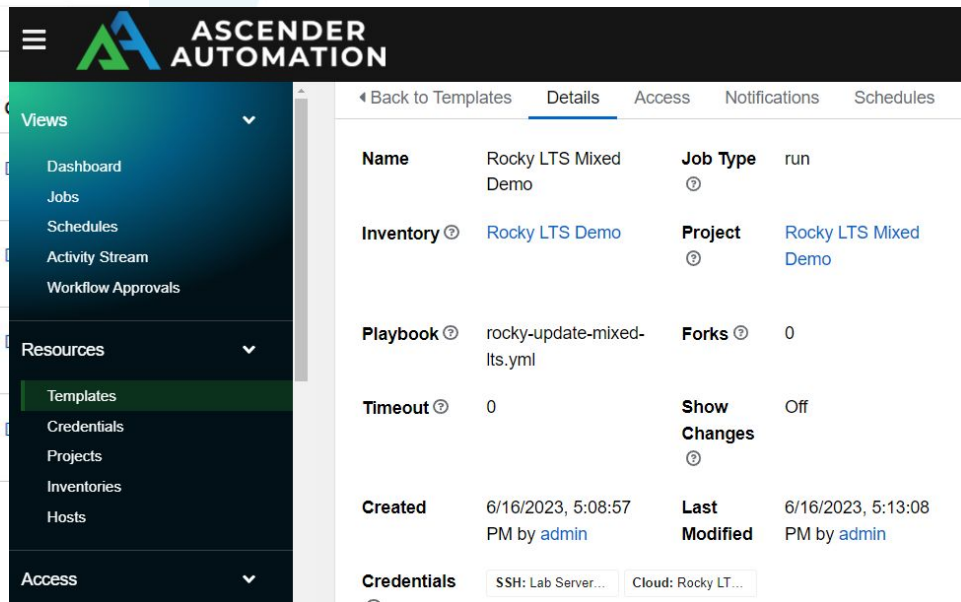
<input type="checkbox"/>	Name	Type
<input type="checkbox"/>	Ansible Galaxy	Ansible Galaxy/Automation Hub API Token
<input type="checkbox"/>	Demo Credential	Machine
<input type="checkbox"/>	Lab Servers Root	Machine
<input type="checkbox"/>	Rocky LTS Mixed Demo Mnt Key	Mountain Access Key
<input type="checkbox"/>	VMWare Server	generic credential 1

# Ascender Job Templates



The screenshot shows the Ascender Automation interface. The top header is black with the Ascender Automation logo (a stylized 'A' in green and blue) and the text 'ASCENDER AUTOMATION'. On the left, there is a dark blue sidebar with a 'Views' section containing links to Dashboard, Jobs, Schedules, Activity Stream, and Workflow Approvals. Below this is a 'Resources' section with links to Templates, Credentials, Projects, Inventories, and Hosts. The 'Templates' link is highlighted. The main content area shows a list of job templates with columns for Name, Type, and a checkbox. The templates listed are 'Demo Job Template', 'Rocky LTS Mixed Demo', 'SNOW Provision VM', and 'SNOW Rocky LTS Mixed Demo'.

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



The screenshot shows the details of a job template in the Ascender Automation interface. The top header is black with the Ascender Automation logo and the text 'ASCENDER AUTOMATION'. Below the header, there is a navigation bar with tabs: Back to Templates, Details (selected), Access, Notifications, and Schedules. The main content area shows the details of the 'Rocky LTS Mixed Demo' job template. The details are organized into sections: Name, Inventory, Playbook, Timeout, Created, Last Modified, and Credentials. The 'Name' section shows 'Rocky LTS Mixed Demo'. The 'Inventory' section shows 'Rocky LTS Demo'. The 'Playbook' section shows 'rocky-update-mixed-lts.yml'. The 'Timeout' section shows '0'. The 'Created' section shows '6/16/2023, 5:08:57 PM by admin'. The 'Last Modified' section shows '6/16/2023, 5:13:08 PM by admin'. The 'Credentials' section shows 'SSH: Lab Server...' and 'Cloud: Rocky LT...'.

Name	Job Type
Rocky LTS Mixed Demo	run

Inventory	Project
Rocky LTS Demo	Rocky LTS Mixed Demo

Playbook	Forks
rocky-update-mixed-lts.yml	0

Timeout	Show Changes
0	Off

Created	Last Modified
6/16/2023, 5:08:57 PM by admin	6/16/2023, 5:13:08 PM by admin

Credentials
SSH: Lab Server... Cloud: Rocky LT...

# Ascender Job Output

ASCENDER  
AUTOMATION

Views

Dashboard

Jobs

Schedules

Activity Stream

Workflow Approvals

Resources

Templates

Credentials

Projects

Inventories

Hosts

Access

Jobs

> ☐ Name     1 - 20 of 82

Name	Status	Type	Start Time	Finish Time
> <input type="checkbox"/> 83 — Cleanup Job Details	Successful	Management Job	7/16/2023, 3:54:50 PM	7/16/2023, 3:54:56 PM
> <input type="checkbox"/> 82 — Cleanup Expired Sessions	Successful	Management Job	7/14/2023, 3:56:30 PM	7/14/2023, 3:56:30 PM
> <input type="checkbox"/> 81 — Cleanup Expired OAuth 2 Tokens	Successful	Management Job	7/14/2023, 3:56:10 PM	7/14/2023, 3:56:10 PM
> <input type="checkbox"/> 80 — Cleanup Activity Stream	Successful	Management Job	7/11/2023, 3:54:48 PM	7/11/2023, 3:54:48 PM

Back to Jobs

Details

Output

Job ID	76	Status	Successful	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

```
0 SSH password:
1
2 PLAY [Perform package update with LTS support] ***** 10:26:04
3
4 TASK [Collect what version of linux is currently running] ***** 10:26:04
5 ok: [greg-rocky86nolts]
6 ok: [greg-rocky86lts]
7
8 TASK [Display the os_version for each server] ***** 10:26:05
9 ok: [greg-rocky86lts] => {
10   "os_version_before.stdout": "PRETTY_NAME=\"Rocky Linux 8.6 (Green Obsidian)\""
11 }
12 ok: [greg-rocky86nolts] => {
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] ***** 10:26:05
17 skipping: [greg-rocky86nolts]
18 ok: [greg-rocky86lts]
```

# Ascender Survey

[Templates](#) > [SNOW Provision VM](#) > [Survey](#)

## Edit Question

Question \*

Name of new VM

Description

Answer type \* ?

Text

☒ Required

Minimum length

0

Maximum length

1024

Answer variable name \* ?

vm\_name

Default answer

NewVM1

### 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

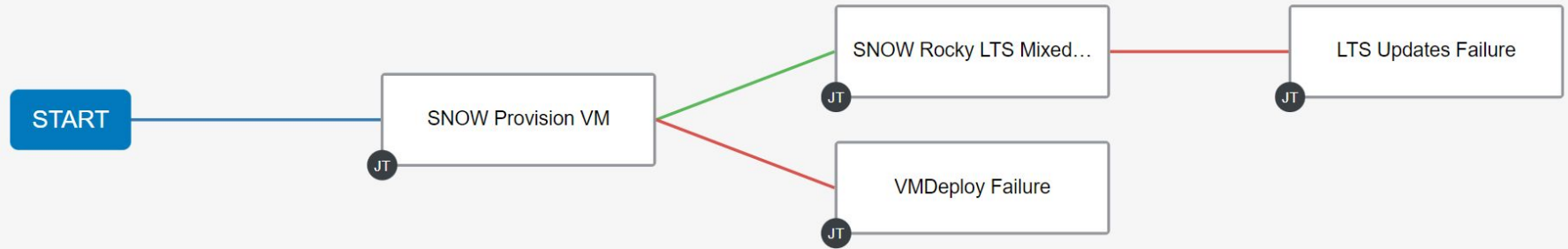
Network subnet mask \*

Next

Back

Cancel

# Ascender Workflow

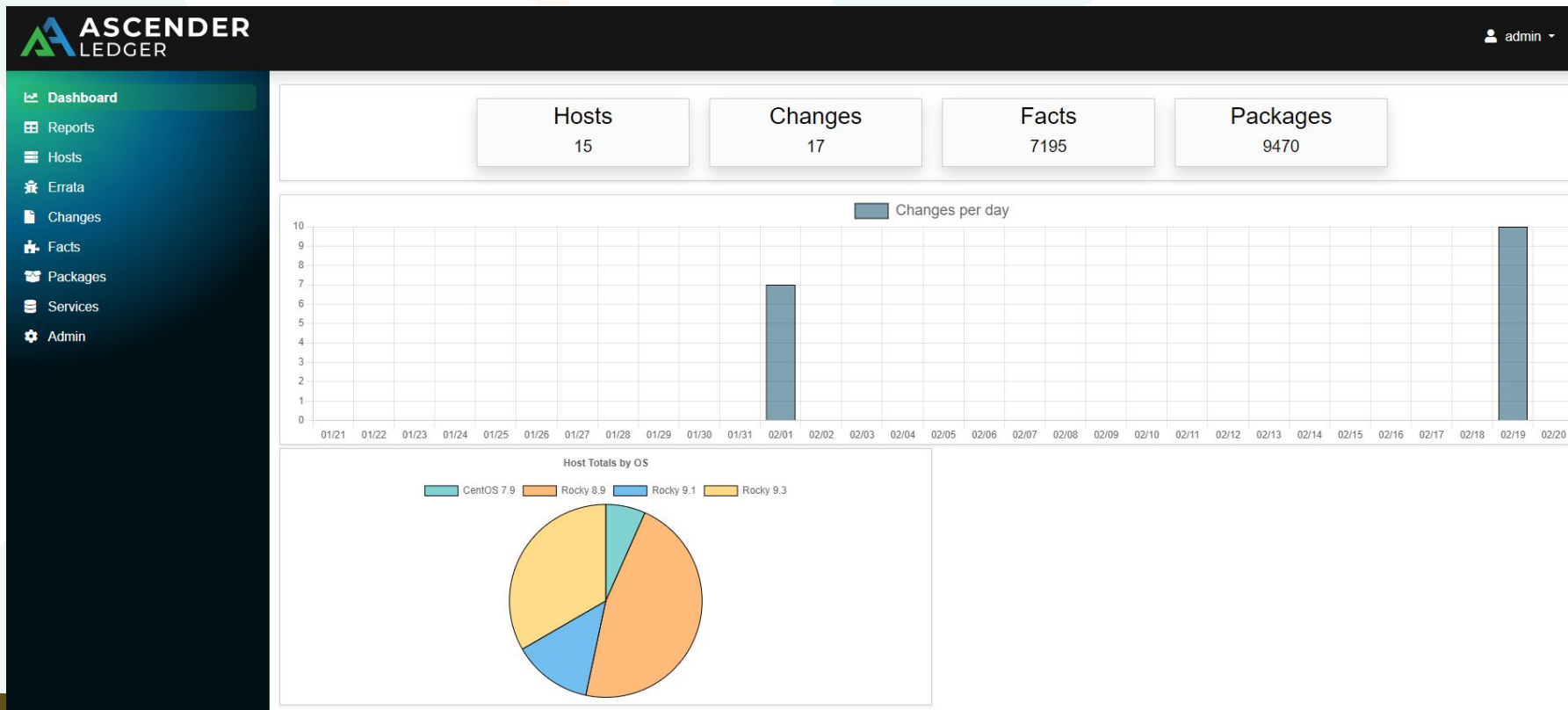


# Ledger/Ledger Pro

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

# Ledger Pro Dashboard



# Ledger Pro Facts

Dashboard

Reports

Hosts

Errata

Changes

**Facts**

Packages

Services

Admin

Hosts

jc-books

×

▼

Facts

Choose a fact...

▼

Types

Choose a type...

▼

Clear

Show

25

▼

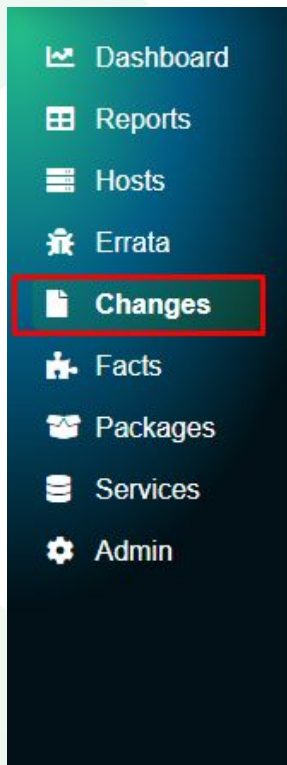
entries

Search:

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



Add a File

Filters:

Show  entries

Search:

	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

# Ledger Pro Reports

Reports

Hosts

Errata

Changes

Facts

Show 25 entries

Name

Linux Servers

Windows Servers

## Linux Servers

Show 50 entries

Copy

CSV

Excel

PDF

Print

Search:

Showing 1 to 15 of 15 entries

Hostname	IP Address	Distro	Distro Version	Python Version	CPUs	Mem
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

Report NameLinux ServersSave

Sort FieldHostnameASCSave

Columns

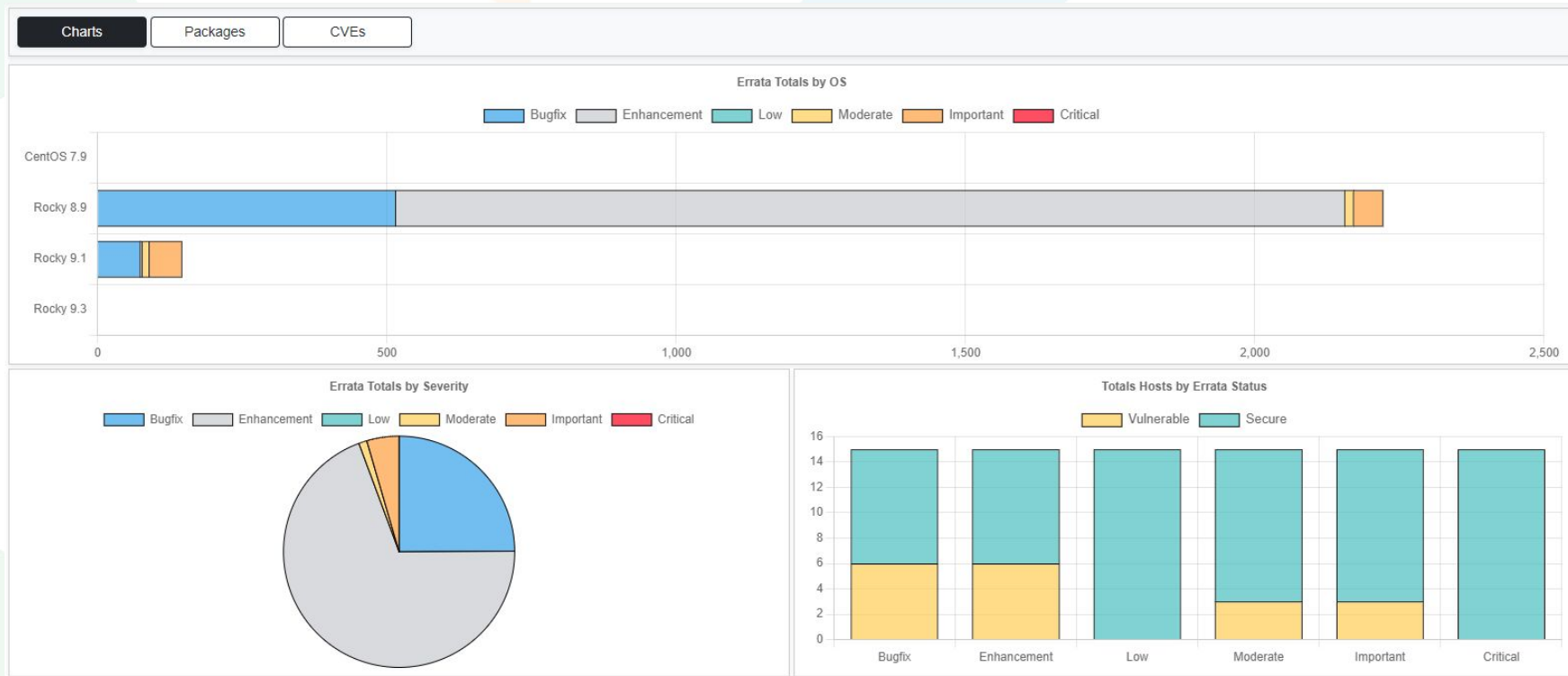
Fact	Display	
ansible_hostname	Hostname	<div></div>
ansible_default_ipv4.address	IP Address	<div></div>
ansible_distribution	Distro	<div></div>

Filters

Fact	Compare	Value
ansible_system	Equals	Linux

Add Filter

# Ledger Pro Errata



# Ledger Pro Service Status

Hosts

Service

State

Status

jc-books x ▾

Choose a service... ▾

Choose a state... ▾

Choose a status... ▾

Clear

Show 25 ▾ entries

Search:

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

CIQ

# Ledger Pro Host Status

jc-demo-rocky9

Host Info

Packages

Services

Errata

## Inventory Hostname (Ascender)

jc-demo-rocky9

## Server FQDN

jc-demo-rocky9.ciqlab.dev

## System Vendor

VMware, Inc.

## Operating System

Rocky Linux 9.1

## Kernel

5.14.0-162.23.1.el9\_1.x86\_64

## IPv4 Addresses

10.0.110.113

## IPv6 Addresses

fe80::250:56ff:fe87:8c3e

## Default IPv4

Interface : ens192  
Address : 10.0.110.113  
Netmask : 255.255.255.0  
Mac Address : 00:50:56:87:8c:3e  
Gateway : 10.0.110.1  
MTU : 1500  
DNS : 10.0.110.1

## Hardware

Memory (Mb) : 3910  
Processors : 2  
Processor Cores : 1  
Total Cores : 2

## SELinux Status

Disabled

## Python

Version : 3.9.14  
Path : /usr/bin/python3.9

## TimeZone

CST / CDT (-0600)

## Disks

### sda

Host : Serial Attached SCSI controller: VMware PVSCSI  
SCSI Controller (rev 02)  
Model : Virtual disk  
Size : 32.00 GB  
Sectors : 67108864 (512)

## Partitions

### sda1 (UUID: C016-F5CD)

Size : 600.00 MB  
Sectors : 1228800 (512)

### sda2 (UUID: 0704dd31-8a6c-4ba9-acca-85e9b6453d90)

Size : 1.00 GB  
Sectors : 2097152 (512)

### sda3 (UUID: )

Size : 30.41 GB  
Sectors : 63778816 (512)

## Mounts

/ (UUID: ed4c915c-8916-4484-a331-22d3a869eef9)  
Size : 27.19 GiB  
/boot (UUID: 0704dd31-8a6c-4ba9-acca-85e9b6453d90)  
Size : 1014 MiB  
/boot/efi (UUID: C016-F5CD)  
Size : 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

**Advisory :** RLSA-2023:3559

**Published :** 2023-06-13

**Severity :** Important

**Updated Package :** c-ares-0:1.17.1-5.el9\_2.1.x86\_64.rpm

**CVEs:**  
[CVE-2023-32067](#)

**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



1.1B Connect To Lab



Mountain



Rocky Linux



Apptainer



Warewulf



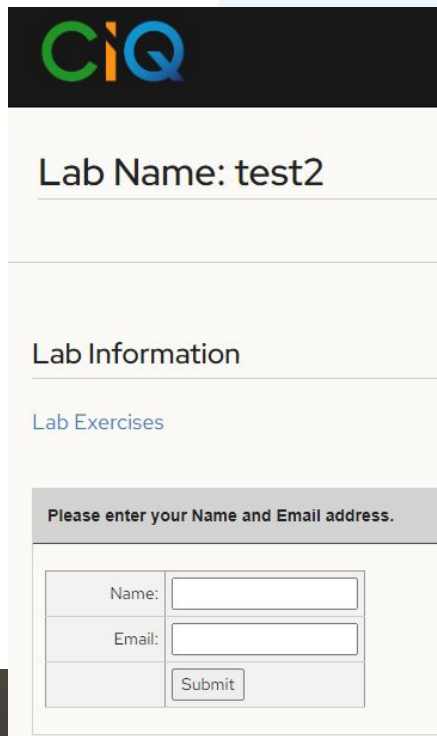
Fuzzball



Ascender

# Browse To: <http://workshop.ciqlab.dev/>

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



The screenshot shows a web interface for CiQ. At the top is the CiQ logo. Below it is a form with the following sections:

- Lab Name:** A text input field containing the text "test2".
- Lab Information:** A section header.
- Lab Exercises:** A section header.
- Please enter your Name and Email address.** A grey instruction bar.
- Form Fields:** Two input fields labeled "Name:" and "Email:". Below these is a "Submit" button.

# Lab Resources

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

## Lab Information

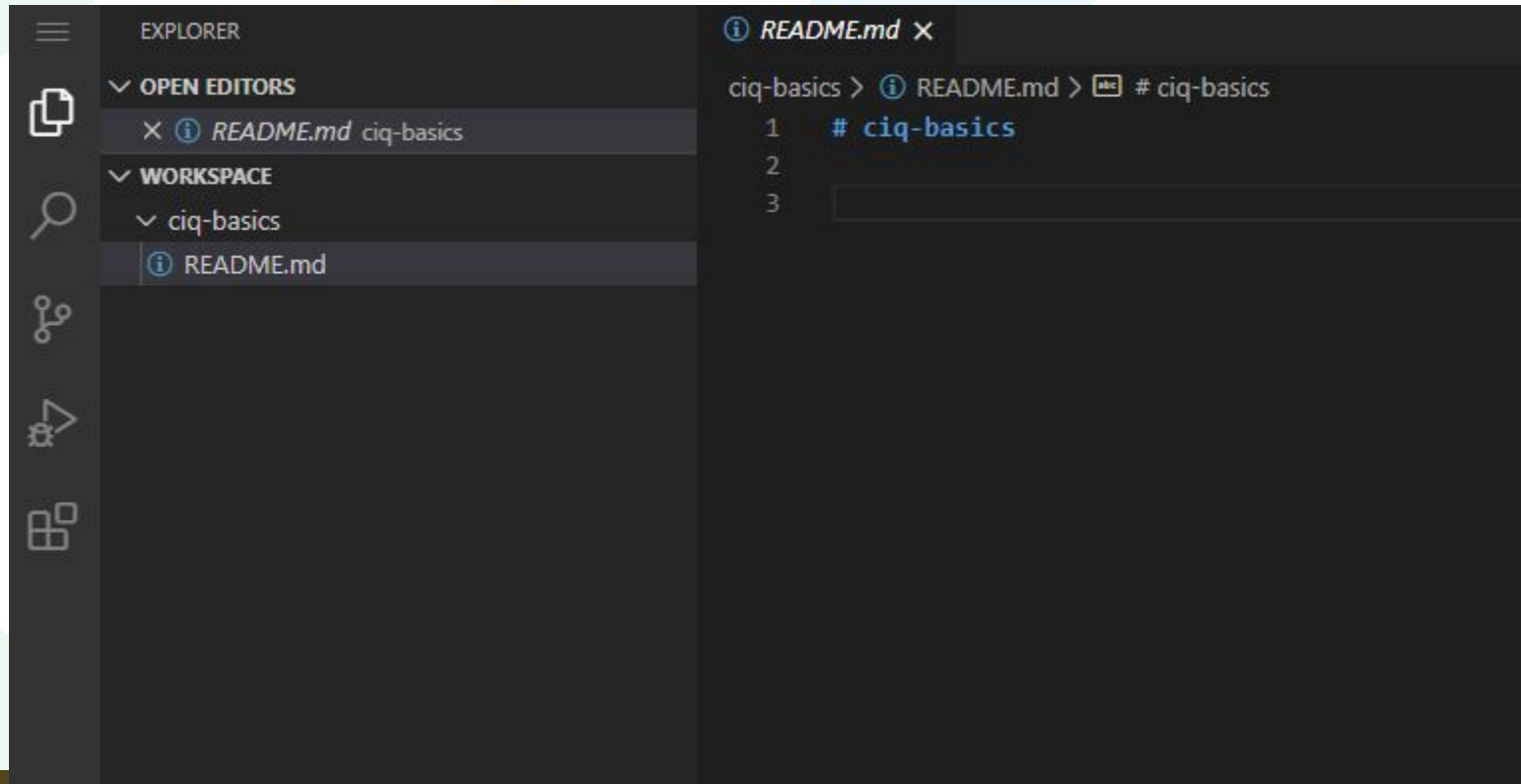
### Lab Exercises

#### Student1

### Lab Nodes

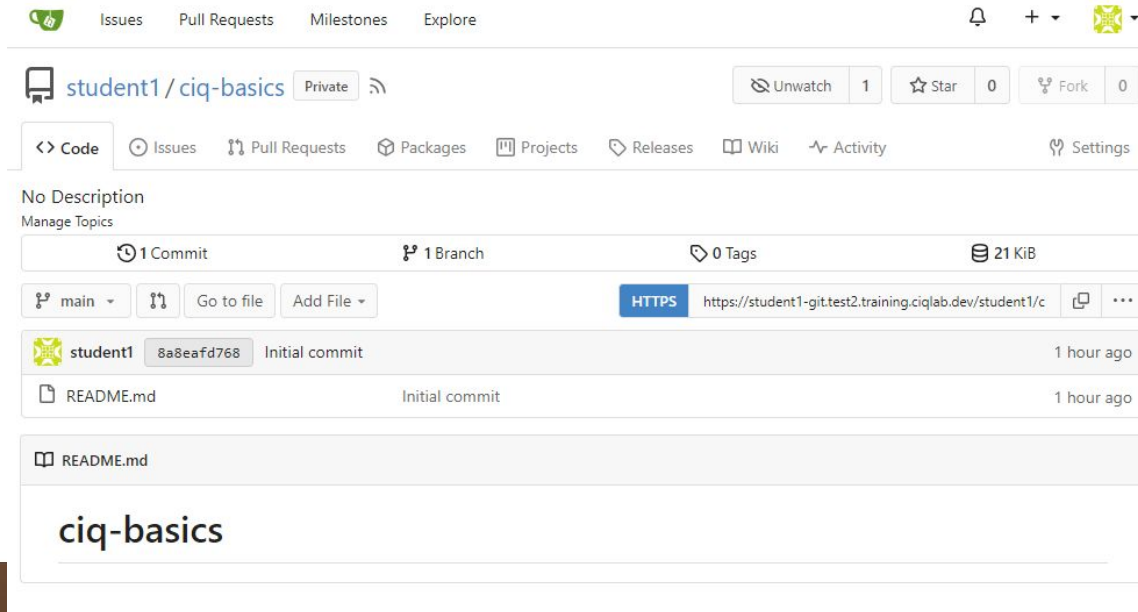
Ascender:	<a href="https://student1.test2.training.ciqlab.dev">https://student1.test2.training.ciqlab.dev</a>
VSCode:	<a href="https://student1-vscode.test2.training.ciqlab.dev">https://student1-vscode.test2.training.ciqlab.dev</a>
GIT:	<a href="https://student1-git.test2.training.ciqlab.dev">https://student1-git.test2.training.ciqlab.dev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

# VSCode Interface



# Gitea Server

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





1.1B LAB - 10 mins



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender



## 1.2 Build a simple playbook



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender



# Writing Playbooks

Playbooks are written in YAML

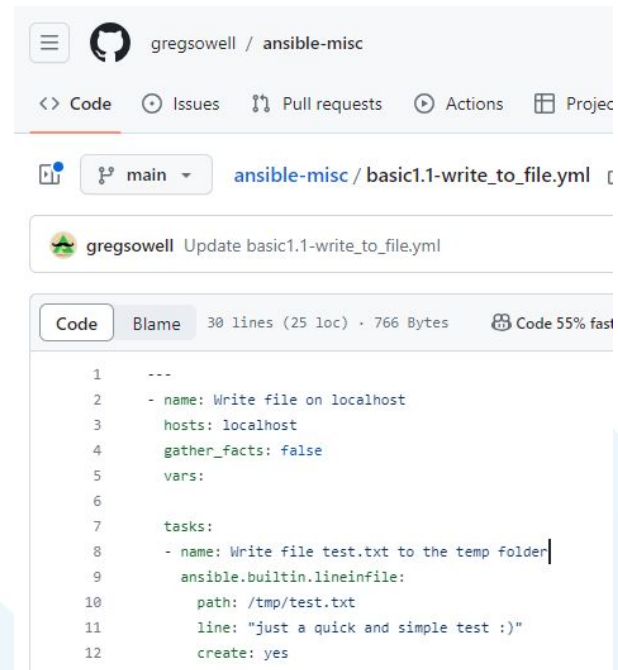
- Human readable
- **Spaces not tabs** - usually 2 spaces

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 displays a GitHub repository named 'gregsowell / ansible-misc'. The file 'basic1.1-write\_to\_file.yml' is selected, showing a commit by 'gregsowell' titled 'Update basic1.1-write\_to\_file.yml'. The code is displayed in a light-themed editor with line numbers 1 through 12. The playbook content includes a 'name' field, 'hosts' set to 'localhost', 'gather\_facts' set to 'false', and a task to write a file to '/tmp/test.txt' with the content 'just a quick and simple test :)'.

```
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: Basics write file  
  hosts: basics-host  
  gather_facts: false  
  
  tasks:  
    - name: Write file test.txt to the temp folder  
      ansible.builtin.lineinfile:  
        path: /tmp/test.txt  
        line: "just a quick and simple test :)"  
        create: yes  
  
    - name: Grab contents of file  
      ansible.builtin.shell: cat /tmp/test.txt  
      register: file_contents  
  
    - name: Display file_contents  
      ansible.builtin.debug:  
        var: file_contents.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: Basics write file  
  hosts: basics-host  
  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: Basics write file  
hosts: basics-host  
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
```



# Ascender Inventory

[Inventories](#) > [Rocky LTS Demo](#)

## Hosts

<a href="#">&lt;</a> <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="text"/>	<a href="#">Add</a>	<a href="#">Run Command</a> <a href="#">Delete</a>
		1 - 2 of 2 <a href="#">v</a>	
Name <a href="#">↑</a>		Description <a href="#">↑</a>	Related Groups
<input type="checkbox"/>	greg-rocky86lts	lts_demo	<input checked="" type="checkbox"/> On
<input type="checkbox"/>	greg-rocky86nonlts	lts_demo	<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: Write file test.txt to the temp folder
  ansible.builtin.lineinfile:
    path: /tmp/test.txt
    line: "just a quick and simple test :)"
    create: yes

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

- name: Display file_contents
  ansible.builtin.debug:
    var: file_contents.stdout_lines
```







1.2 Lab - 35 mins



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender



## 1.3 Ascender Job Template Creation

# Credentials

The screenshot displays the Ascender Automation web interface. The top navigation bar is dark with the logo on the left and user information on the right. A left sidebar contains a menu with 'Views' and 'Resources' sections. The 'Credentials' page is active, showing a table of credentials. The table has columns for Name, Type, and Actions. Three credentials are listed: 'Ansible Galaxy', 'basics-credential', and 'scm-basics-course'. The 'basics-credential' is highlighted. At the bottom right, there is a pagination control showing '1 of 1 page'.

**ASCENDER AUTOMATION**

student1

**Views**

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

**Resources**

- Templates
- Credentials**
- Projects
- Inventories
- Hosts

## Credentials


☐ Name    1 - 3 of 3




<input type="checkbox"/>	Name	Type	Actions
<input type="checkbox"/>	Ansible Galaxy	Ansible Galaxy/Automation Hub API Token	
<input type="checkbox"/>	basics-credential	Machine	
<input type="checkbox"/>	scm-basics-course	Source Control	

1 - 3 of 3 items 1 of 1 page

# Projects

≡

 **ASCENDER  
AUTOMATION**

   student1

Views

Dashboard

Jobs

Schedules

Activity Stream

Workflow Approvals

Resources

Templates

Credentials

Projects


Inventories

## Projects

>

☐

Name ▾








Add

Delete

1 - 1 of 1 ▾

<

>

	Name ↑	Status	Type	Revision	Actions
>	<input type="checkbox"/> basics-course	 Successful	Git	b295ca2 	  

1 - 1 of 1 items ▾



<< <




1

> >>

of 1 page

# Inventories

**ASCENDER  
AUTOMATION**

 student1

Views

Dashboard

Jobs

Schedules

Activity Stream

Workflow Approvals

Resources


Templates

Credentials

Projects

Inventories




## Inventories

☐ Name  

Add

Delete



1 - 1 of 1




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

1 - 1 of 1 items

1 of 1 page

# Templates

**ASCENDER  
AUTOMATION**

 student1

**Views** ▾

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


**Resources** ▾

- Templates**
- Credentials
- Projects

## Templates

> ☐

Name ▾









Add ▾

Delete

1 - 9 of 9 ▾

<

>

	Name ↑	Activity	Last Ran ↓	Type ↓	Actions
> <input type="checkbox"/>	basics-conditionals	✓✓✓	2/26/2024, 3:54:28 PM	Job Template	  
> <input type="checkbox"/>	basics-handlers	✓❗❗	2/26/2024, 4:26:18 PM	Job Template	  



1.3 Lab - 20 mins



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender



## 1.4 Execute a playbook



Mountain



Rocky Linux



Apptainer



Warewulf



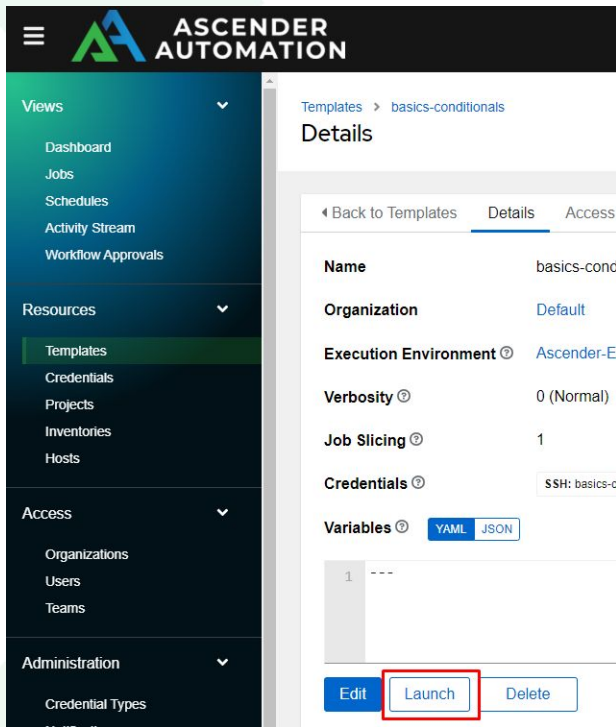
Fuzzball



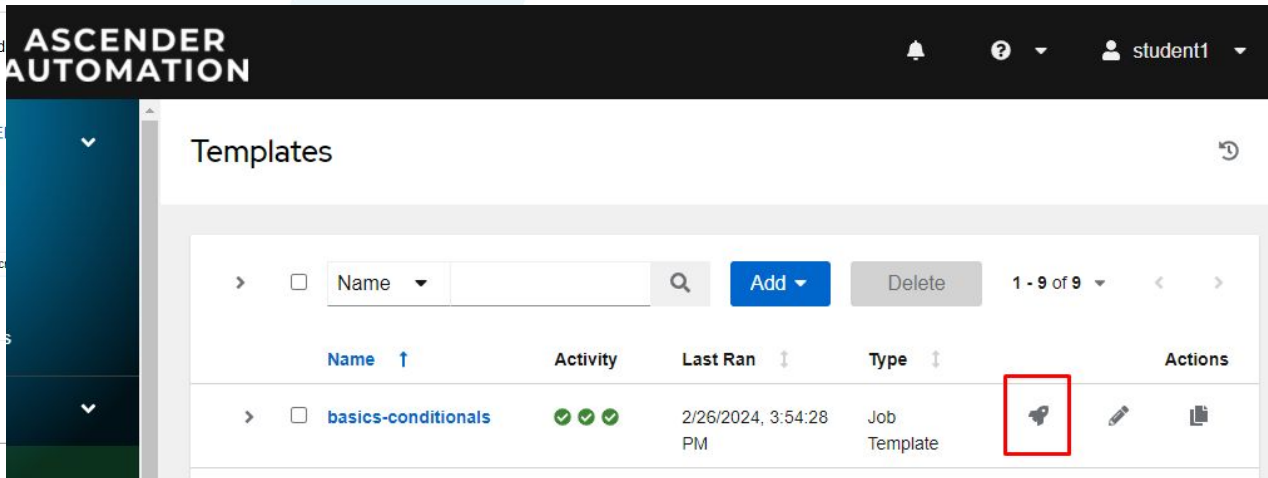
Ascender






# Launch Ansible Playbook



The sidebar menu for Ascender Automation is displayed. It includes sections for Views, Resources, Access, and Administration. The 'Resources' section is expanded, showing 'Templates' as the selected item. The 'Launch' button in the bottom right of the sidebar is highlighted with a red box.



The 'Templates' page in Ascender Automation is shown. It features a table with columns for Name, Activity, Last Ran, Type, and Actions. The 'basics-conditionals' template is highlighted. The 'Launch' button in the Actions column is highlighted with a red box.

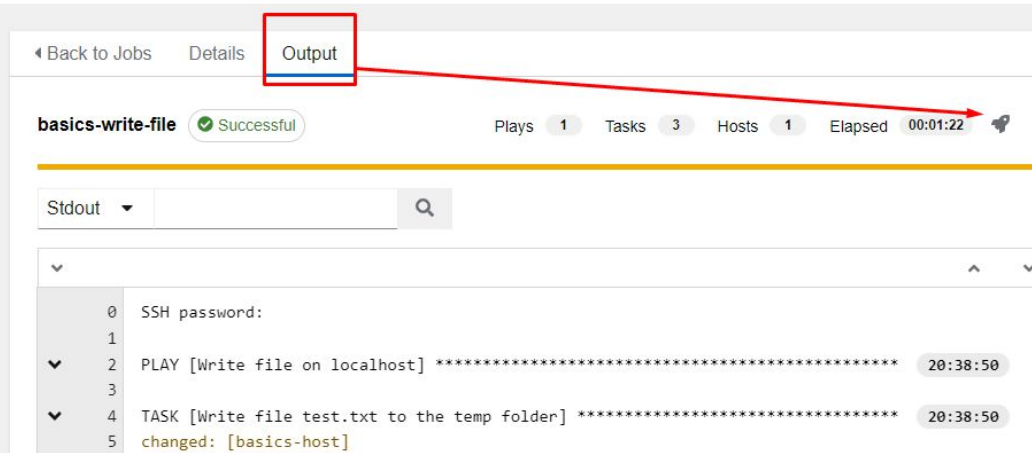
Name	Activity	Last Ran	Type	Actions
basics-conditionals	✓✓✓	2/26/2024, 3:54:28 PM	Job Template	  

# Idempotence

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

Jobs > 13 - basics-write-file

## Output



Jobs > 13 - basics-write-file

Output

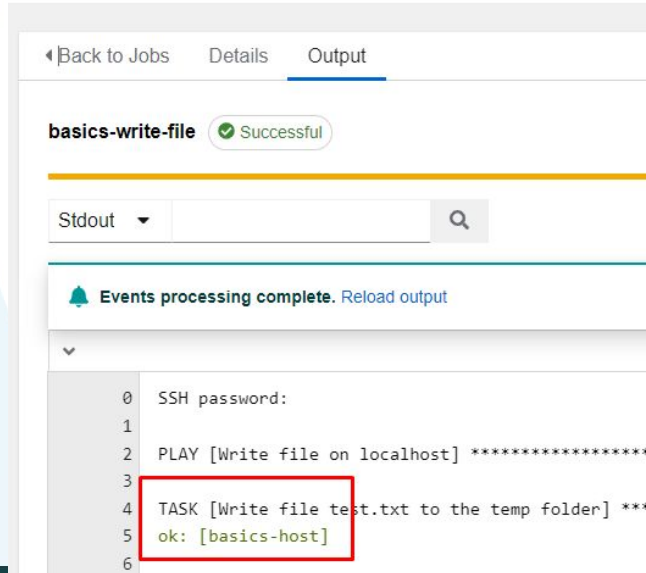
basics-write-file Successful Plays 1 Tasks 3 Hosts 1 Elapsed 00:01:22

Stdout

```
0 SSH password:
1
2 PLAY [Write file on localhost] ***** 20:38:50
3
4 TASK [Write file test.txt to the temp folder] ***** 20:38:50
5 changed: [basics-host]
```

Jobs > 15 - basics-write-file

## Output



Jobs > 15 - basics-write-file

Output

basics-write-file Successful

Stdout

Events processing complete. [Reload output](#)

```
0 SSH password:
1
2 PLAY [Write file on localhost] *****
3
4 TASK [Write file test.txt to the temp folder] ***
5 ok: [basics-host]
6
```

# Output

ok	Everything executed fine, but no changes were made.
changed	Everything ran fine, but it appears that a change was made.
failed	The task failed for a host. This will cause the playbook to show a failed status.
skipping	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 Variables



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/Dictionaryes

- 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** JSON

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





1.5 Lab - 20 mins



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender



## 1.6 Conditionals And Blocks



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender

# 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"
```

# Blocks

- Group together **multiple tasks**
- Allow for error recovery via “rescue”
- Allow one **conditional** to affect whole block

```
vars:
  tacos: 5
  day: Tuesday

tasks:
  - name: taco block
    when: tacos >= 5 and day == "Tuesday"
    block:
      - name: Print taco time
        ansible.builtin.debug:
          msg: "we have plenty of tacos"
      - name: Print taco date
        ansible.builtin.debug:
          msg: "And it's Tuesday!"
```



1.6 Lab - 15 mins



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender



## 1.7 Handlers



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender



# Restarting Services

Standard procedure for restarting services

```
---  
- name: Install Web Server  
  hosts: web  
  
  tasks:  
    - name: Install Apache  
      ansible.builtin.dnf:  
        name: httpd  
        state: latest  
        register: httpd_install  
  
    - name: Restart httpd  
      ansible.builtin.service:  
        name: httpd  
        state=restarted  
        when: httpd_install.changed
```

# Restarting Services Cont

## Multiple restart conditions

tasks:

- name: Install Apache

  - ansible.builtin.dnf:

    - name: httpd

    - state: latest

  - register: httpd\_install

- name: Restart httpd

  - ansible.builtin.service:

    - name: httpd

    - state=restarted

  - when: httpd\_install.changed

- name: Apache alternate port

  - lineinfile: dest=/etc/apache2/ports.conf

    - regexp="^Listen "

    - line="Listen {{ http\_port }}"

    - state=present

  - register: alt\_port

- name: Restart httpd

  - ansible.builtin.service:

    - name: httpd

    - state: restarted

  - when: alt\_port.changed

~~~~~

# Handlers

- “**notify**” option used to signal handler
- **handler** section keeps handlers
- Called only once even if signaled by multiple tasks

## tasks:

- name: Install Apache  
ansible.builtin.dnf:  
  name: httpd  
  state: latest  
  register: httpd\_install  
  **notify**: Restart httpd
- name: Apache alternate port  
lineinfile: dest=/etc/apache2/ports.conf  
  regexp="^\s\*Listen "  
  line="Listen {{ http\_port }}"  
  state=present  
  register: alt\_port  
  **notify**: Restart httpd

## handlers:

- name: Restart httpd  
ansible.builtin.service:  
  name: httpd  
  state=restarted



1.7 Lab - 15 mins



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender



1.8 Loops



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender

# 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 }}"
```



1.8 Lab - 20 mins



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender



1.9 Templates



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender

# Template Basics

- Jinja2 templating engine
- Text files
- End in .j2 extension
- `{{ }}` is used for variable replacement, no `""` needed
- Provides
  - Variable replacement
  - Looping
  - Etc.
- Called via `ansible.builtin.template` task or lookup plugin

## Task

```
- name: Write template file
  ansible.builtin.template:
    src: templates/test_file.txt.j2
    dest: /tmp/test_file.txt
  vars:
    purpose: Ansible Testing
```

## Template

This server is {{ inventory\_hostname }}.  
It's purpose is {{ purpose }}.

## Resulting File

This server is ansible-test1.  
It's purpose is Ansible Testing.

# Template Advanced

- `{% %}` is used for things like if/then/else and loops
- `{# #}` is used for comments within the template

# Template Advanced

## Playbook

```
---  
- name: Write file on localhost  
  hosts: localhost  
  gather_facts: false  
  vars:  
    fav_foods:  
      - tacos  
      - burgers  
      - donuts  
  
  tasks:  
    - name: Write file test.txt to the temp folder  
      ansible.builtin.template:  
        dest: /tmp/food.txt  
        src: templates/food.txt.j2
```

## food.txt.j2

```
My favorite foods are:  
{% for item in fav_foods %}  
  -<[ {{ item }} ]>-  
{% endfor %}
```



1.9 Lab - 15 mins



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender





## 1.10 Roles



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender

# Role Basics

- Makes playbook portions reusable
- Like a function or subroutine
- Just YAML files in a folder structure

# Role Make Up

- Defaults - maintains default variables. Lowest level of precedence
- Files - maintains any additional files
- Handlers - handlers used by role
- Meta - metadata. Includes references to other roles
- Tasks - all of the tasks and task files
- Templates - any jinja2 templates used
- Tests - test playbooks, etc.
- Vars - variables that most often aren't overwritten

```
[root@localhost ~]# tree test-role/  
test-role/  
├── defaults  
│   └── main.yml  
├── files  
├── handlers  
│   └── main.yml  
├── meta  
│   └── main.yml  
├── README.md  
├── tasks  
│   └── main.yml  
├── templates  
├── tests  
│   ├── inventory  
│   └── test.yml  
└── vars  
    └── main.yml
```

# Example 1

```
---  
- name: Write file on localhost  
  hosts: localhost  
  gather_facts: false  
  
  tasks:  
    - name: Write file test.txt to the temp folder  
      ansible.builtin.lineinfile:  
        path: /tmp/test.txt  
        line: "just a quick and simple test :)"  
        create: yes  
  
    - name: Grab contents of file  
      ansible.builtin.shell: cat /tmp/test.txt  
      register: file_contents  
  
roles:  
  - add_users
```

# Ansible Order Of Execution

1. pre\_tasks
2. roles - roles specified in the “roles” section
3. tasks
4. post\_tasks

```
---  
- name: Write file on localhost  
  hosts: localhost  
  gather_facts: false  
  
  pre_tasks:  
    # first stuff here  
  
  roles:  
    - add_users  
  
  tasks:  
    - name: Write file test.txt to the temp folder  
      ansible.builtin.lineinfile:  
        path: /tmp/test.txt  
        line: "just a quick and simple test :)"  
        create: yes  
  
  post_tasks:  
    # last stuff here
```

## Example 2

```
---  
- name: Write file on localhost  
  hosts: localhost  
  gather_facts: false  
  
  tasks:  
    - name: Write file test.txt to the temp folder  
      ansible.builtin.lineinfile:  
        path: /tmp/test.txt  
        line: "just a quick and simple test :)"  
        create: yes  
  
    - name: Grab contents of file  
      ansible.builtin.shell: cat /tmp/test.txt  
      register: file_contents  
  
    - name: Run add_users role  
      include_role:  
        name: add_user
```

# Development

- Create a standard playbook that “works” and includes the parts you want in role
- Cut and paste the role components to role files
- **Optional** Use “ansible-galaxy init role-name” to create the base folder structure for a new role



1.10 Lab - 25 mins



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender





## 2.1 Surveys



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender

# 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

Templates > aaa-test

Survey

◀ Back to Templates

Details

Access

Notifications

Schedules

Jobs

Survey



Add

Delete

## Add Question

Question \*

Description

Answer variable name \* ?

Answer type \* ?

Text ▼

☒ Required

Minimum length

Maximum length

Default answer

# Surveys Question Order

- Click Edit Order and drag the questions around

☐ Add Edit Order Delete ☒ Survey Enabled

| 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        |         |

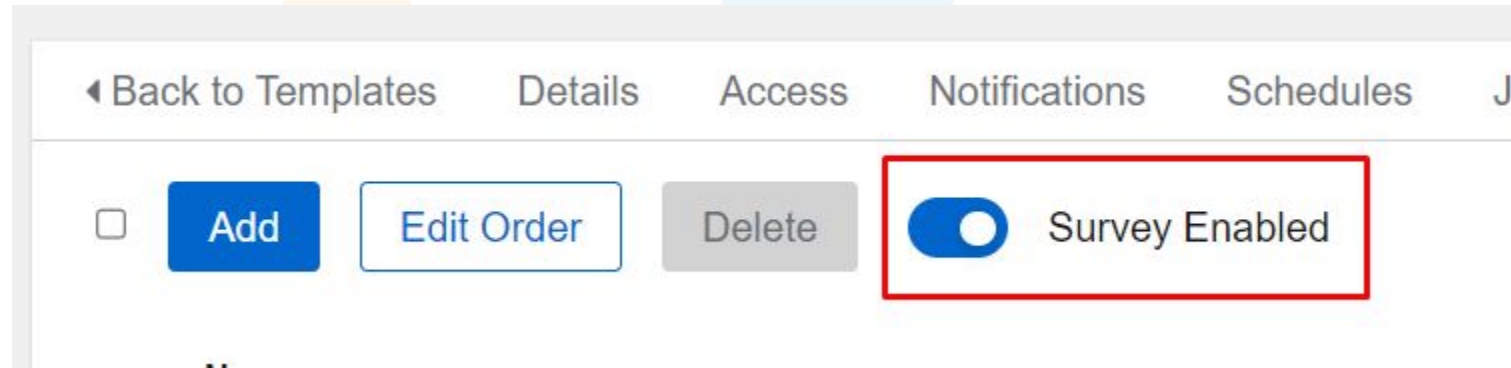
## Survey Question Order

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

Job Slicing 1

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



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender



## 2.2 Workflows



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball

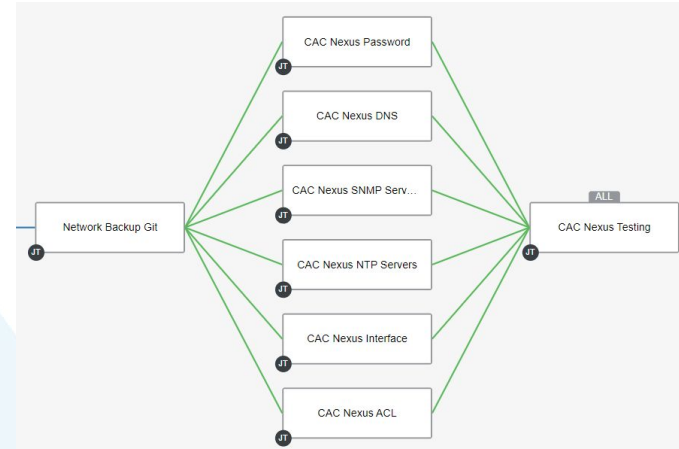
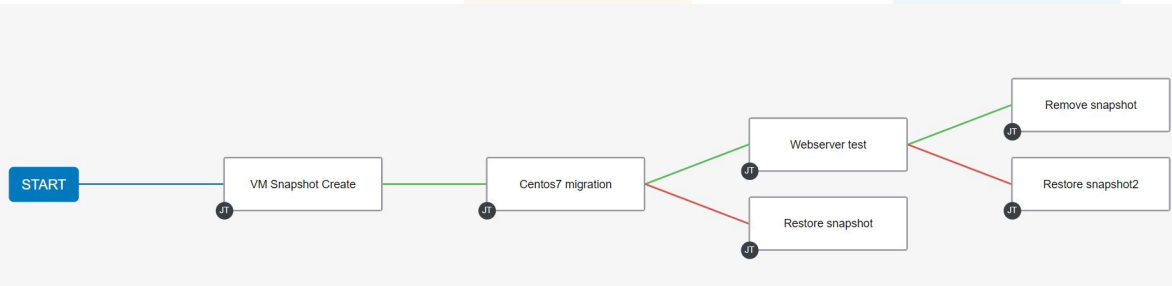


Ascender



# 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

The screenshot displays the Ascender Automation web application. The left sidebar contains a 'Views' section with links to Dashboard, Jobs, Schedules, Activity Stream, and Workflow Approvals, and a 'Resources' section with a link to Templates. The 'Templates' link is highlighted with a red box. The main content area is titled 'Templates' and shows a table with columns for Name, Add, and Last. A red arrow points from the 'Templates' link in the sidebar to the 'Add' button in the table. Another red arrow points from the 'Add' button to the 'Add workflow template' option in the dropdown menu.

ASCENDER  
AUTOMATION

Views

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

Resources

- Templates

Templates

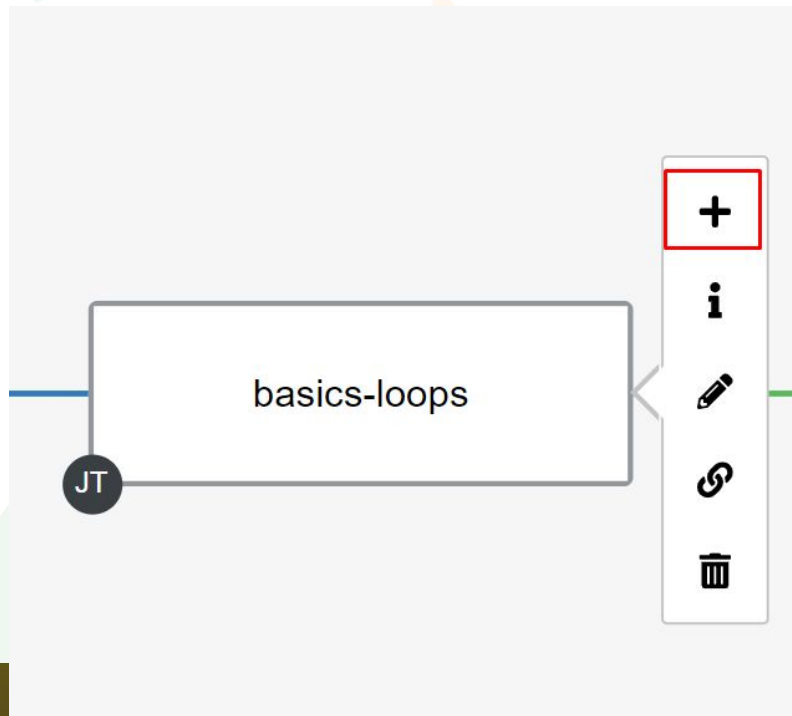
| > | <input type="checkbox"/> | Name     | ▼ | Q | Add | ▼ | De    |
|---|--------------------------|----------|---|---|-----|---|-------|
|   |                          | Name     | ↑ |   |     |   | Last  |
| > | <input type="checkbox"/> | aaa-test |   |   |     |   | 10/25 |

Add job template

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

- 1 Run type

- 2 Node type

### Node Type

Job Template

Name ▾

Name ↑

- ☐ aaa-test
- ☐ Ascender CAC Migration Export
- ☐ Ascender CAC Migration Import
- ☐ basics-loops
- ☐ basics-write-file-vars-survey

Convergence \* ?

Any



2.2 Lab - 20 mins



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender



## 2.3 Role-Based Access Control(RBAC)

# RBAC Basics

Role-Based Access Control used to delegate permissions in the system:  
admin, use, update, read, execute

- **Organization**
  - Top level
  - Collection of all objects in Ascender: users, teams, credentials, jobs, etc.
- **User**
  - User accounts for users/systems to log in to Ascender
  - Can have roles assigned directly
- **Team**
  - A group users can be added to
  - Any roles applied to teams are inherited by their users

User authentication can be local or via remote authentication system



2.3 Lab - 20 mins



Mountain



Rocky Linux



Apptainer



Warewulf



Fuzzball



Ascender



info@ciq.com - (800) 220-5243

**Good luck, keep learning!**