AWX 201: Advanced Automation Techniques with the Ansible AWX Platform

CISCO Live !

Tim Glen
Security Solutions Engineer

Let's Encrypt

HashiCorp Vault

Cisco Webex App

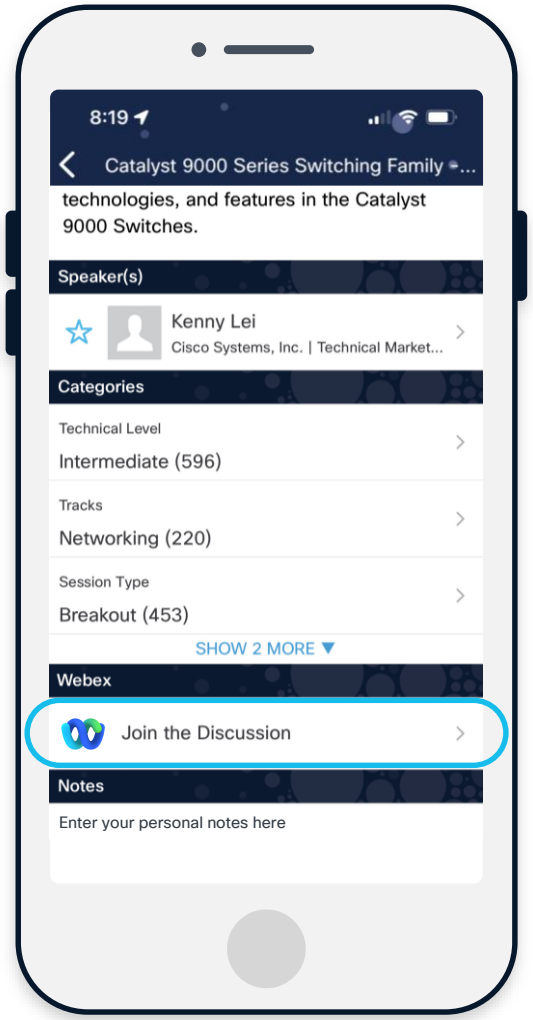
Questions?

Use Cisco Webex App to chat
with the speaker after the session

How

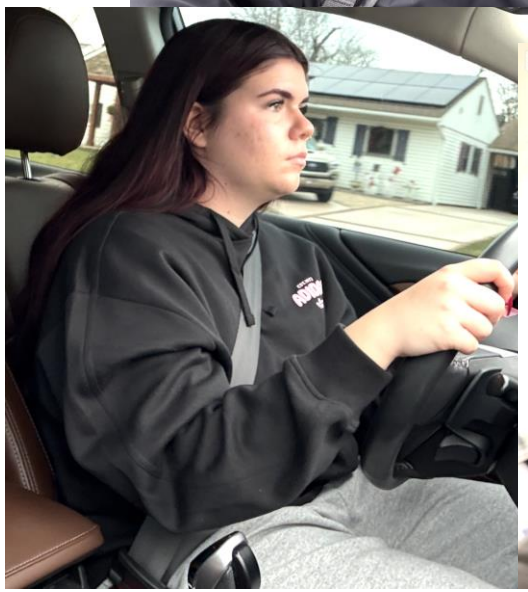
1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space

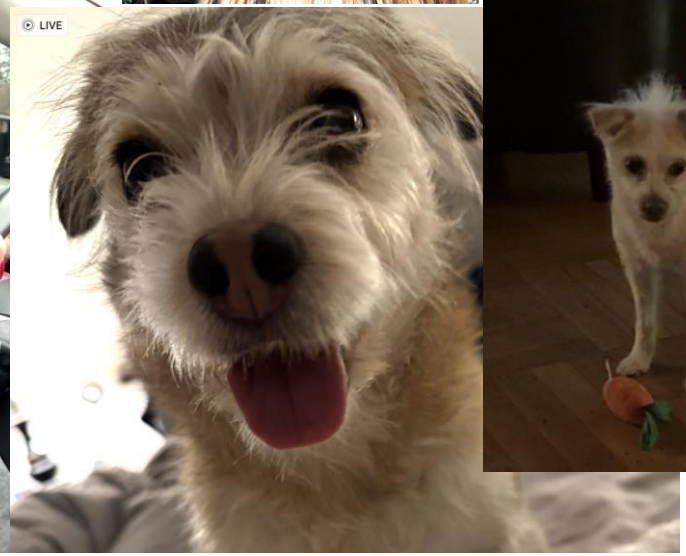
Webex spaces will be moderated by the speaker until June 13, 2025.



Tim (Personal)

- Human
- Husband
- Father
- Dog Dad
- Outside
- Biking
- Driving
- Travel





Tim (Professional)

- Started in IT in 1995, Telephone Tech Support
- Worked 23 years at Web Hosting Provider
  - Managed all routers, switches, firewalls, wireless, security
- Worked at Cisco 6 years
  - Security Systems Engineer

github.com/timmayg

linkedin.com/in/timglen

cs.co/TimGlen

Agenda

1 Introduction

2 Execution Environments

3 Custom Credentials with HashiCorp Vault

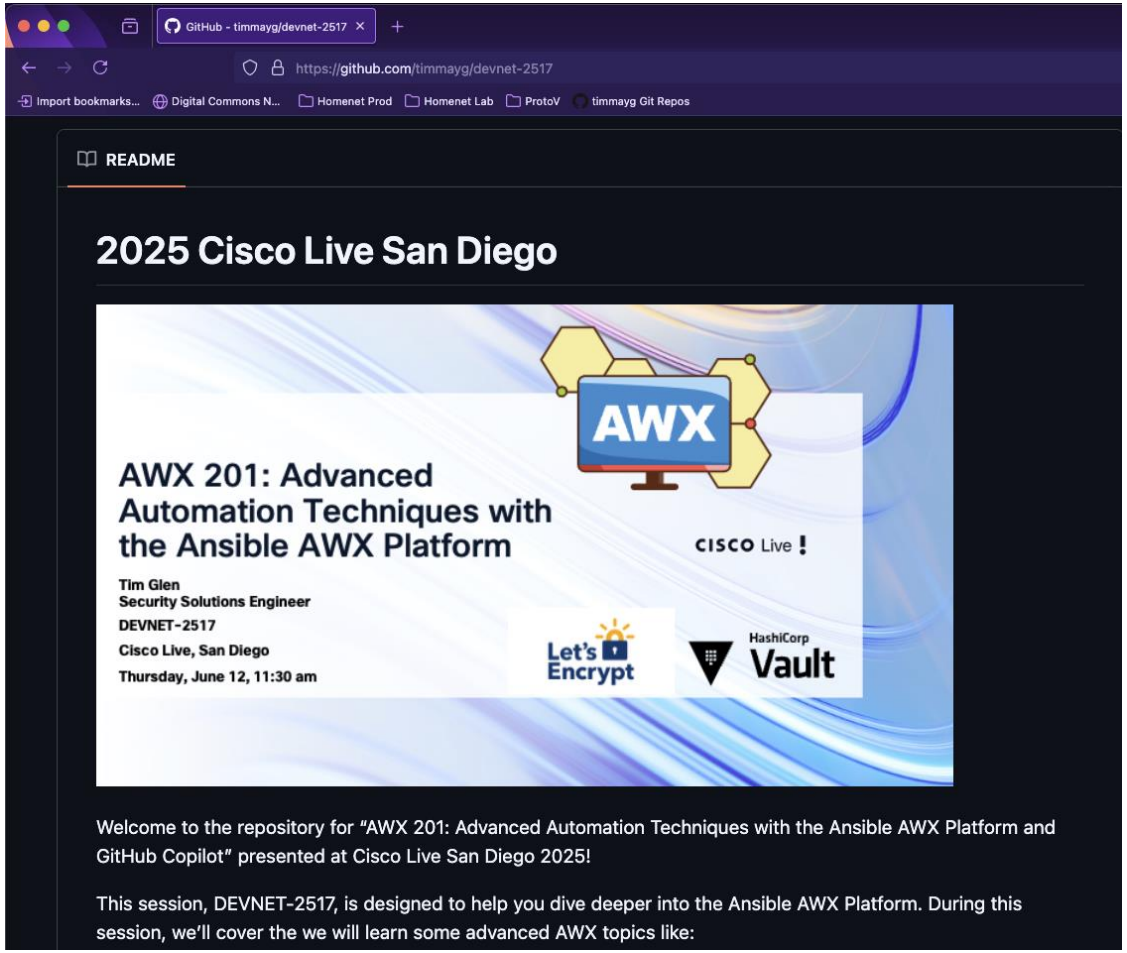
4 Certificate Automation

5 Conclusion

Check Here for Updates

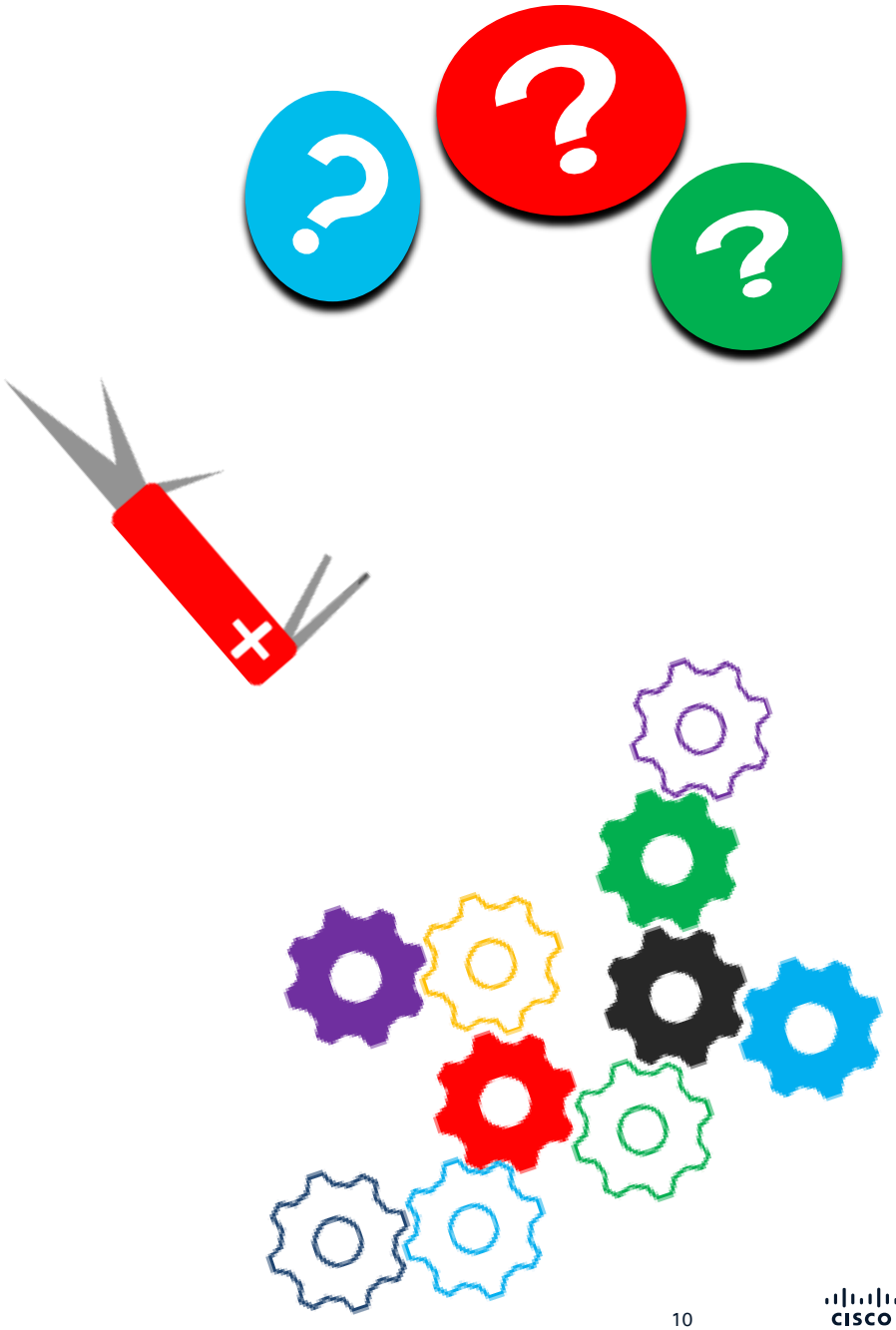
GitHub

https://github.com/timmayg/devnet-2517



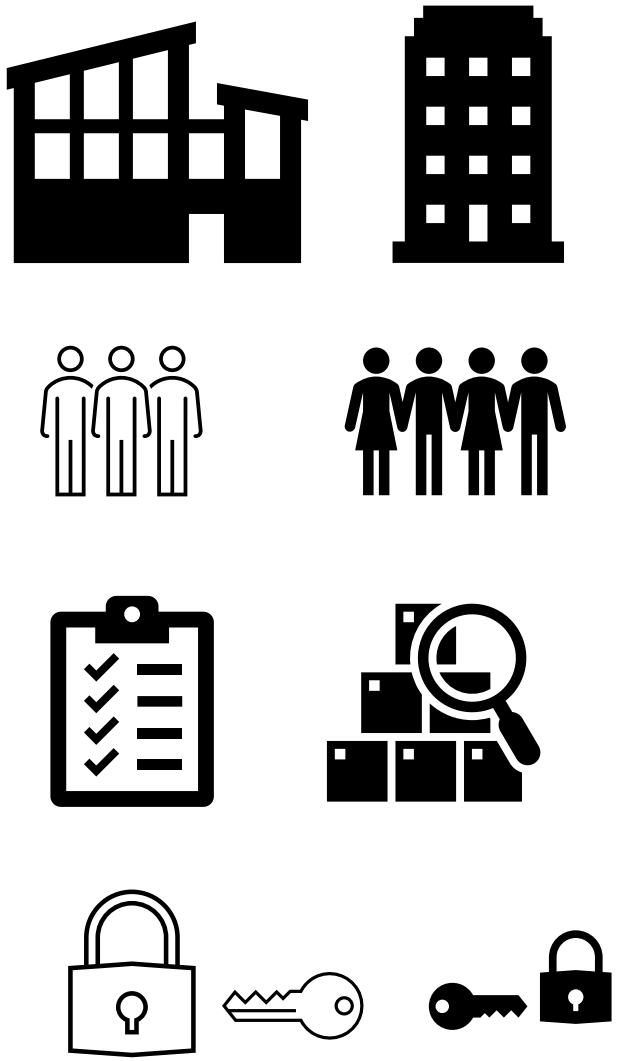
Why Ansible, Why AWX

- Ansible is tool of choice
- Flexibility – Manages Everything
- Easy to write playbooks
- AWX is the next step
- Powerful task engine!
- AWX Web UI & REST API
- Scheduler, Logging



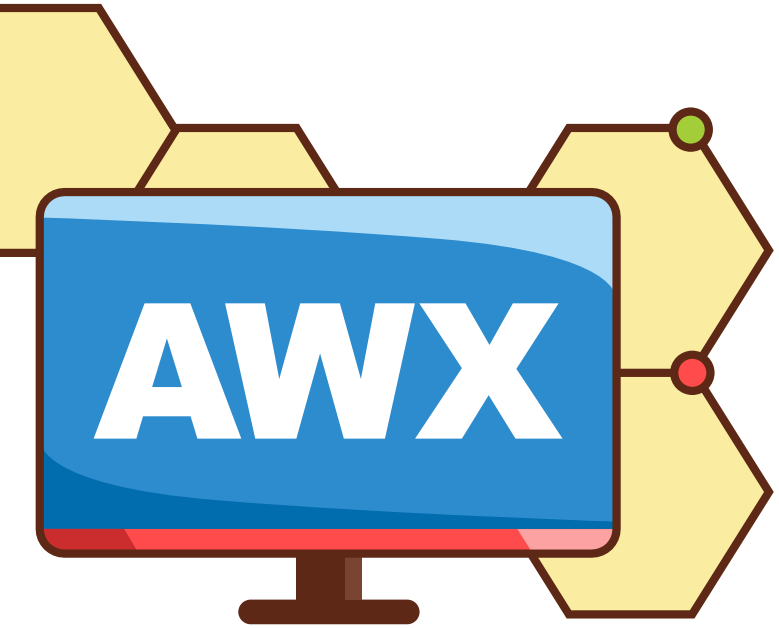
AWX – High Level

- Ansible Automation Platform
- Orgs, Teams & Users
- Templates & Projects
- Hosts & Inventories
- Credentials & Credential Types
- Instances, scaling
- Execution Environments



AWX Integrations & Compatibility

AWX Integrations & Compatibility



HashiCorp Vault

CISCO

ACME

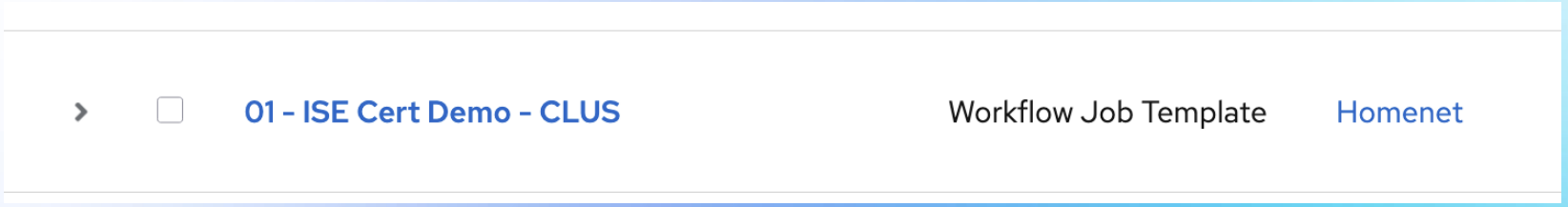
ISE

Let's Encrypt

webex by CISCO

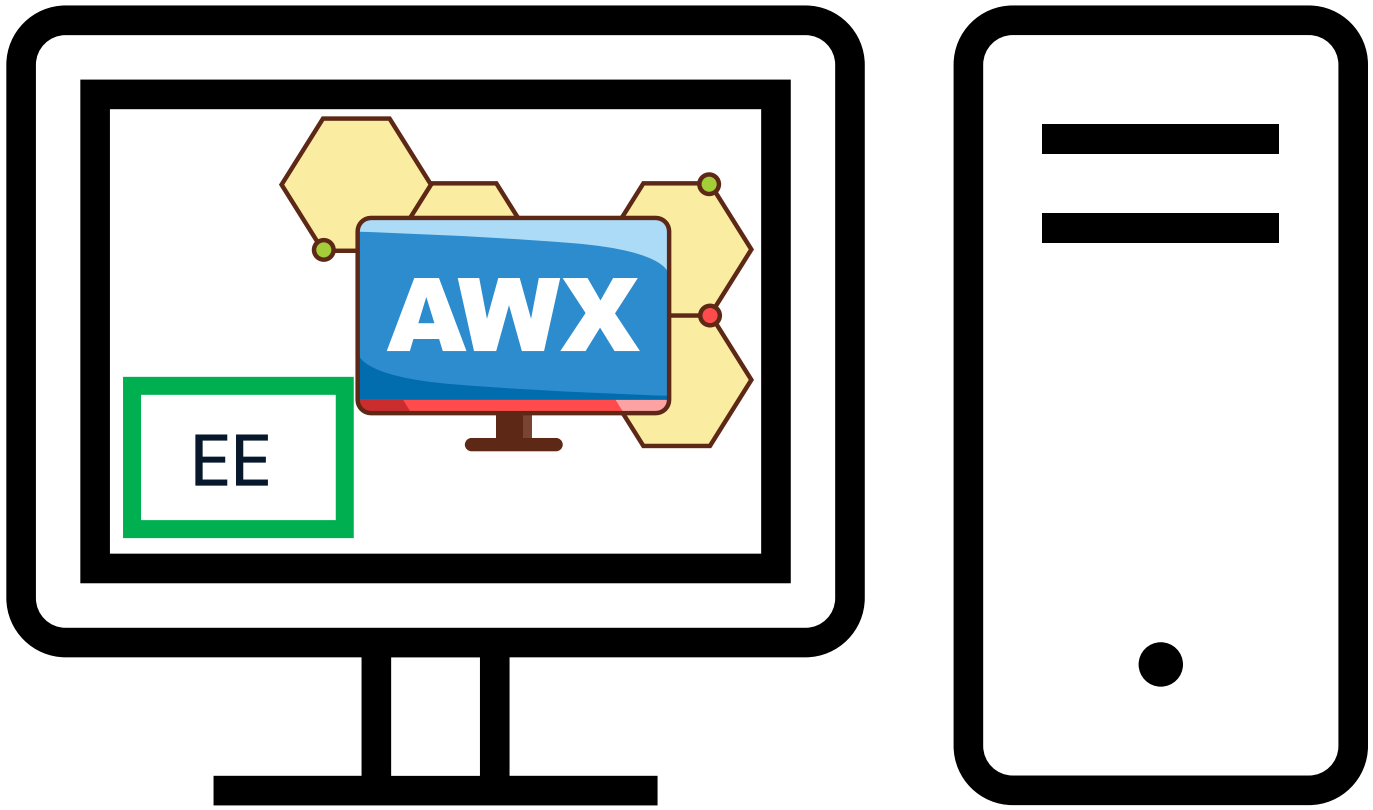
Run ISE Cert Job RIGHT NOW

Label: demo



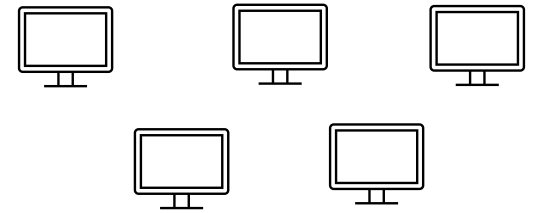
Execution Environments

Execution Environment



- Playbooks \ templates do not run on AWX host OS
- Playbooks run inside of specialized EE container
- AWX server stays clean
- EE gets python libraries, modules etc.

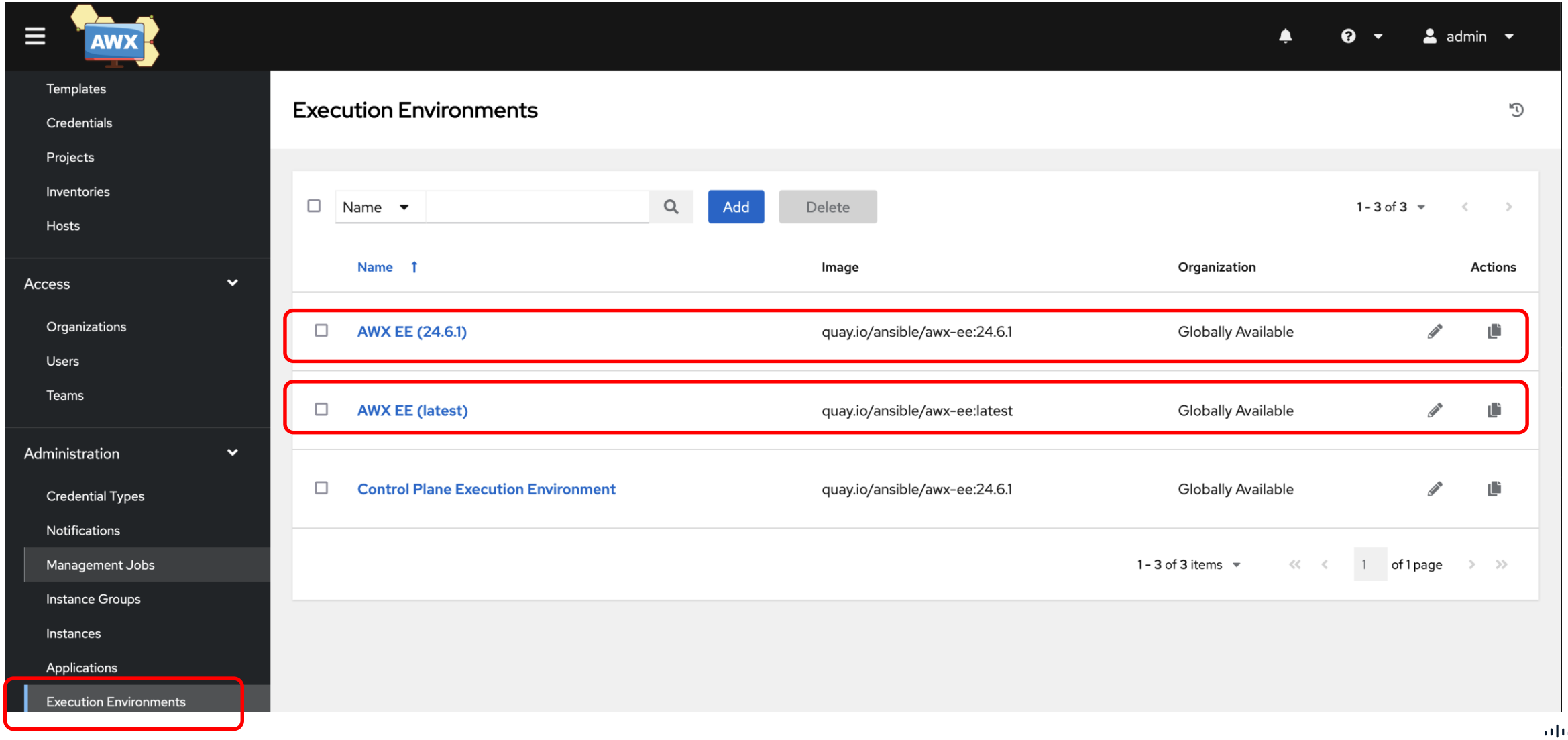
Execution Environment



- Linux container, spun up by AWX at playbook runtime
- Functional Execution Environment is built into AWX – awx-ee
- Build your own!
- Includes ansible-core, ansible-runner, python, python library & package, ansible collections, system dependencies
- Container provides consistency, and scalability, assuring playbooks run
- Stored Container Registries, Docker Hub, Quay IO, private repos

https://docs.ansible.com/ansible/latest/getting_started_ee/index.html

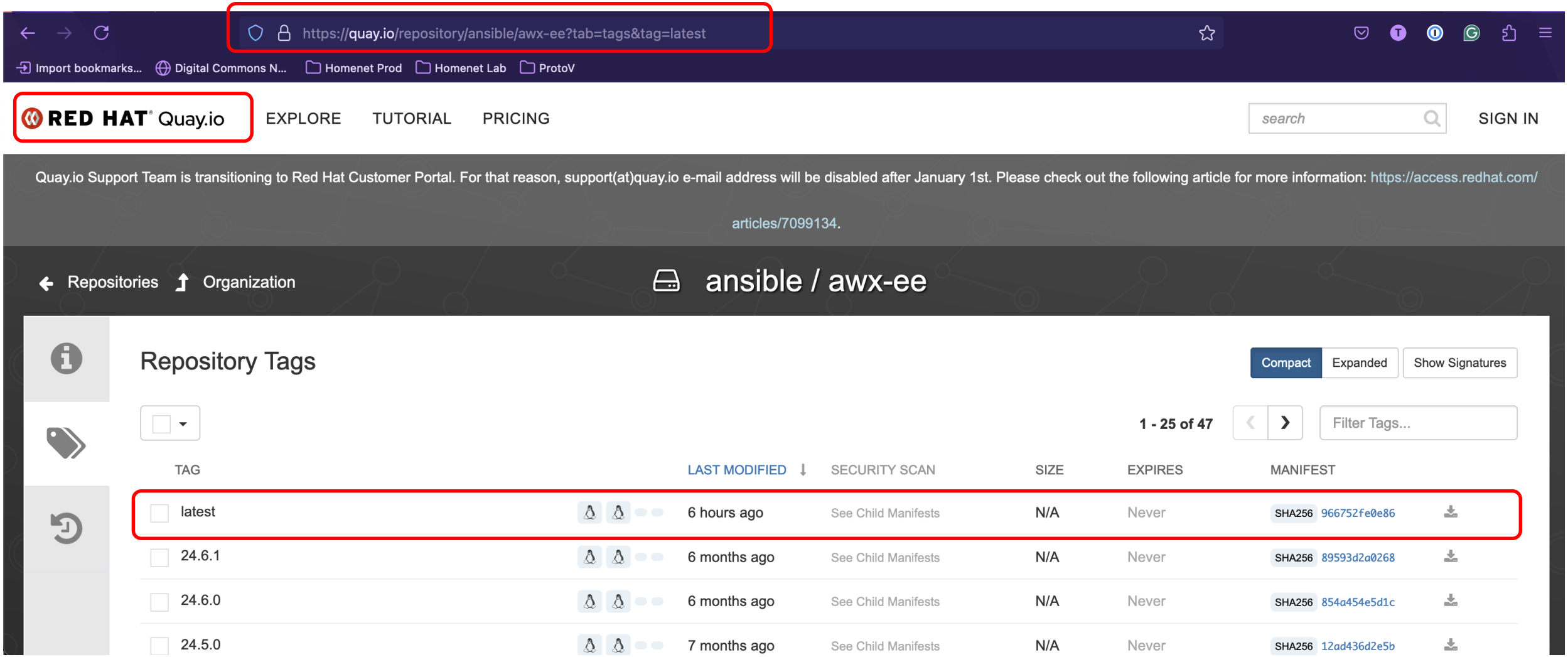
Execution Environment



AWX-EE

Red Hat Quay

Project QUAY



cisco

Updating the Execution Environment

- Update one or more of the following files
  - execution-environment.yaml  **1** Required –base OS, build template
  - bindep.txt  **2** Optional – system-level dependencies
  - requirements.txt  **3** Optional – python packages
  - requirements.yaml  **4** Optional – Ansible Galaxy
- Use ansible-builder to build a new Image
- Verify Image contents using podman
- Upload Image to Container Registry
- AWX downloads the latest image next playbook run

https://developers.redhat.com/articles/2023/05/08/how-create-execution-environments-using-ansible-builder

execution-environment.yaml

- Execution Environment Schema Definition

execution-environment.yaml 1

Users > tiglen > Library > CloudStorage > OneDrive-Cisco > git > timmay-prod > execution-environment.yaml

```yaml
version: 3

images:
  base_image:
    name: quay.io/ansible/awx-ee:latest

dependencies:
  system: bindep.txt
  python: requirements.txt

additional_build_steps:
  prepend_base:
    - RUN yum -y update && yum install -y ftp
  append_base:
    - ENV timmay_prod_version=v1.6
    - RUN ln -sf /usr/share/zoneinfo/America/New_York /etc/localtime
    - RUN alternatives --install /usr/bin/python3 python3 /usr/bin/python3.11 0
    - RUN curl -o /runner/ansible.cfg https://raw.githubusercontent.com/timmayg/timmay-prod/main/ansible.cfg
    - RUN curl -o /runner/release-notes.json https://raw.githubusercontent.com/timmayg/timmay-prod/main/release-notes.json
    - ENV ANSIBLE_CONFIG=/runner/ansible.cfg
    - RUN curl https://packages.microsoft.com/config/rhel/7/prod.repo > /etc/yum.repos.d/microsoft.repo
    - RUN yum install -y powershell
    - RUN pip install ansible-builder

```

Base Image

System - OS Level
Python Package Req's

Additional Steps

Execution Environment Base Image Options

awx-ee - the default

quay.io/ansible/awx-ee

ee-minimal-rhel8

registry.redhat.io/ansible-automation-platform/ee-minimal-rhel8

CentOS stream

quay.io/centos/centos:stream9

- Others too!
- Why ???
- Rebuilding using awx-ee takes > 20 minutes
- Rebuilding & Launching can be faster with lighter

bindep.txt

- OS Level Requirements

requirements.txt

- Python Package requirements

requirements.yaml

- Ansible Galaxy Collections

requirements.yaml 2

Users > tiglen > Library > CloudStorage > OneDrive-Cisco > git > timmay-prod > requirements.yaml

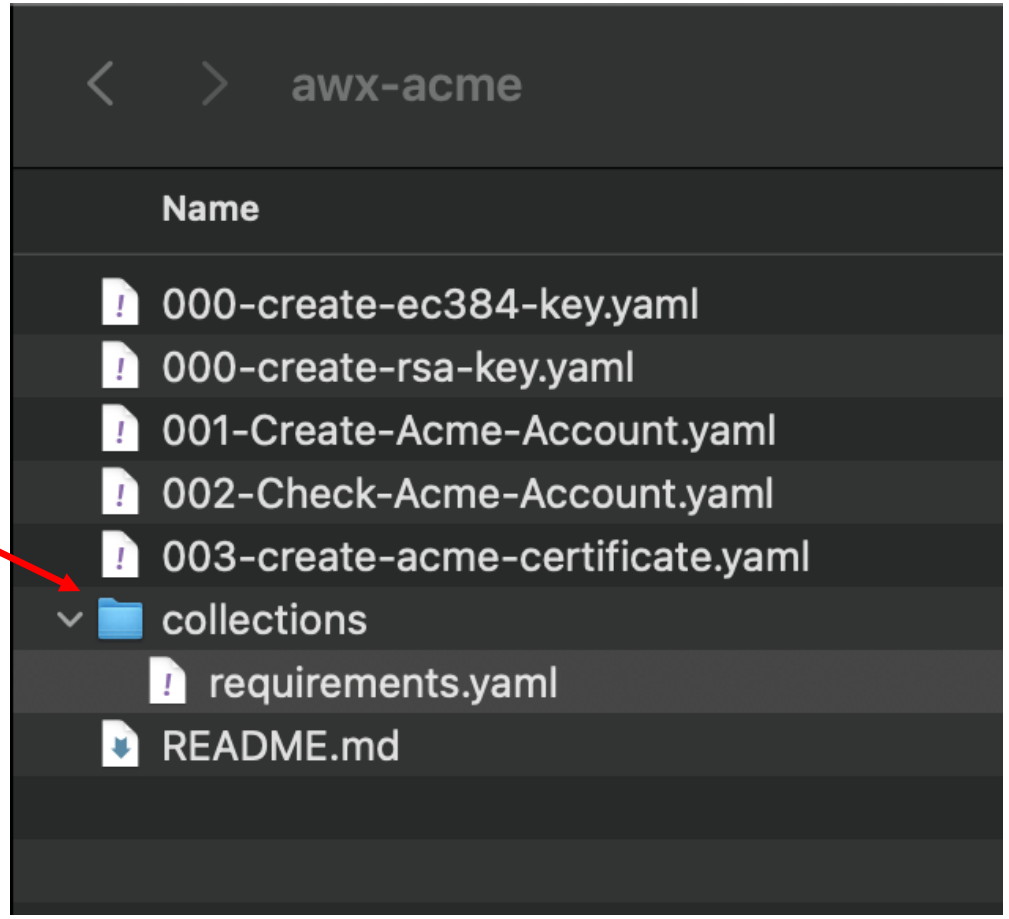
```yaml
---
collections:
  - name: ansible.netcommon
  - name: ansible.utils
  - name: community.crypto
  - name: community.general
  - name: cisco.ios
  - name: cisco.ise
  - name: cisco.fmcansible.fmc_configuration

```

Collections in EE vs Collections in Projects

- Collections in EE are present for any \ all playbooks that are run
- This is speedy

or

- Collections are specified in collections/requirements.yaml
  - This is specified in the Project
  - Every time you Sync an AWX Project the Galaxy Collections need to be downloaded
  - This slows syncing down



Ansible Builder

The `build` command

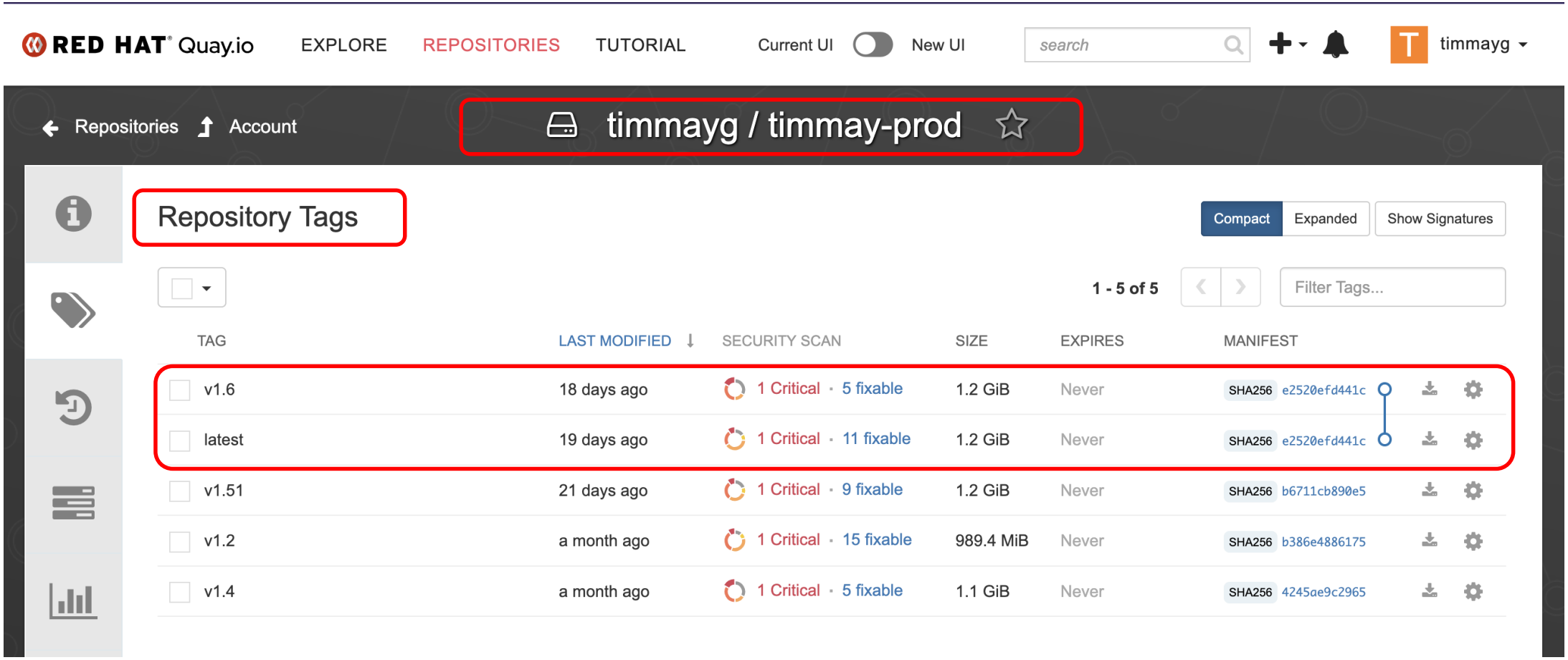
The `ansible-builder build` command:

- takes an execution environment definition file as an input,
- outputs a build instruction file (Containerfile for Podman, Dockerfile for Docker),
- creates a build context necessary for building an execution environment image,
- builds the image.

By default, it looks for a file named `execution-environment.yml` (or `execution-environment.yaml`) in the current directory.

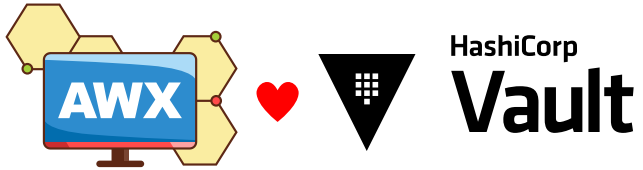
https://ansible.readthedocs.io/projects/builder/en/latest/

Upload your EE to a Container Registry



AWX Custom Credentials
&
External Secret Management
&
Hashi Corp Vault

Secrets Managers

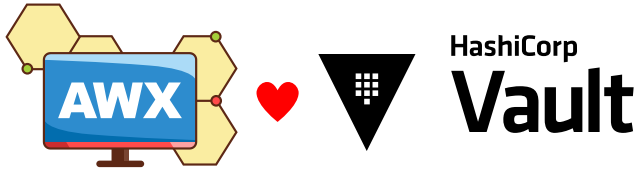


- AWX = Excellent Task Engine
- HashiCorp Vault = Excellent Secrets Management
- AWS Secrets Manager = Excellent Secrets Management
- Azure Key Vault = Excellent Secrets Management

Challenges with AWX built-in Credentials

- Machine Cred Challenges
- Familiarity but limited options

Lessons Learned
Limitation of Built-in Credentials




```yaml
---
- name: Get a List of the ISE Nodes v1.0
  hosts: localhost
  gather_facts: false

  tasks:

    - name: 01 - Read a ISE Credentials from Vault
      community.hashi_vault.vault_kv2_get:
        path: "ise1_credentials"
        url: "https://vault.theglens.net:8200"
        engine_mount_point: "kv"
        auth_method: token
        token: "{{ ansible_password }}"
      register: ise_creds


    - name: 02 - Get a List of the ISE Nodes
      cisco.ise.node_info:
        ise_hostname: "{{ ise_creds.data.data.ise_hostname }}"
        ise_username: "{{ ise_creds.data.data.ise_username }}"
        ise_password: "{{ ise_creds.data.data.ise_password }}"
        ise_verify: true
        ise_debug: false
      register: ise_node_list
      timeout: 120
```

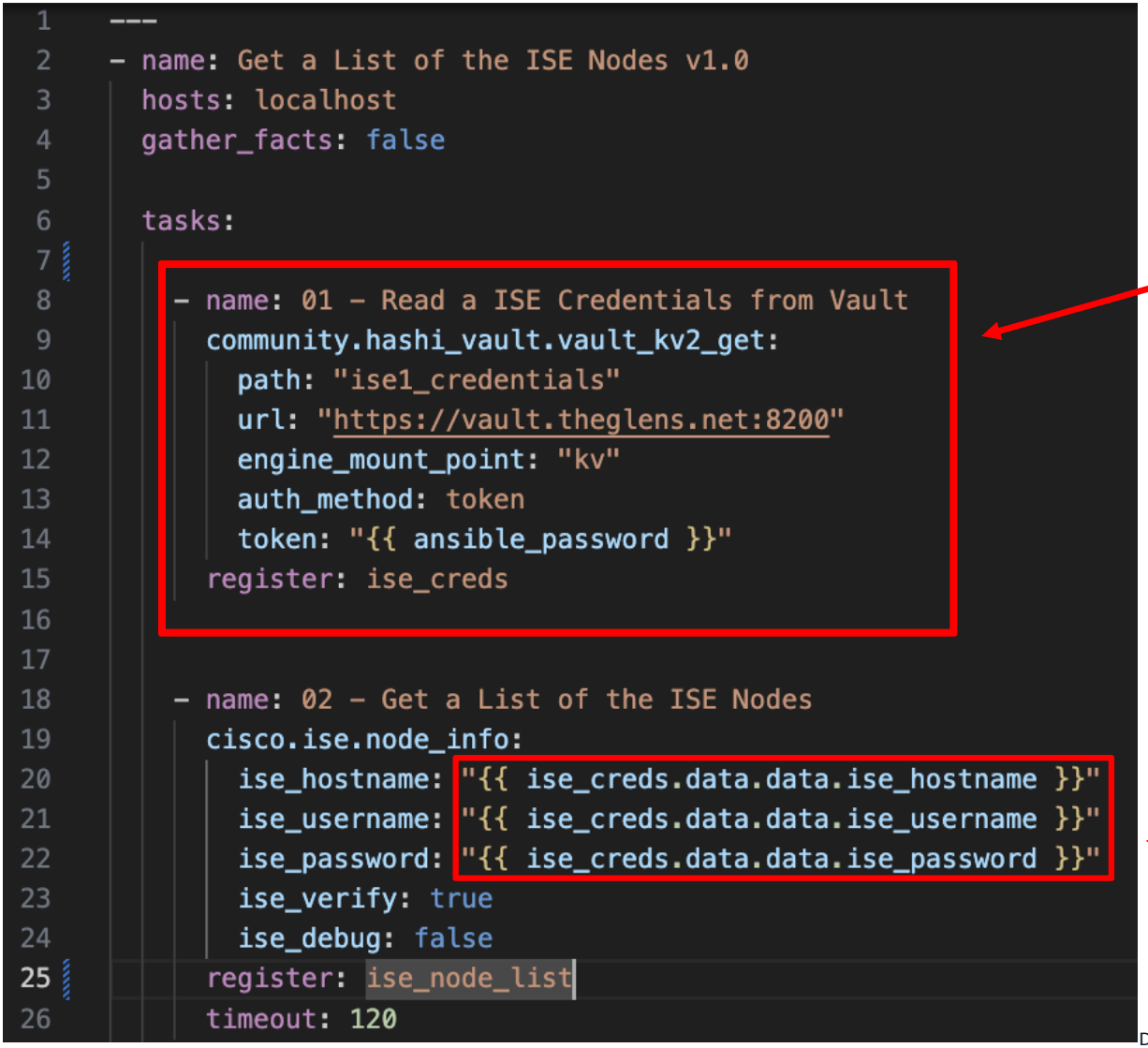
We need to run ansible playbooks to perform ISE Tasks.

Retrieve ISE User, Pass & hostname from Vault

Each ISE task requires authentication

cisco

Lessons Learned
Limitation of Built-in Credentials



```yaml
---
- name: Get a List of the ISE Nodes v1.0
  hosts: localhost
  gather_facts: false

  tasks:

    - name: 01 - Read a ISE Credentials from Vault
      community.hashi_vault.vault_kv2_get:
        path: "ise1_credentials"
        url: "https://vault.theglens.net:8200"
        engine_mount_point: "kv"
        auth_method: token
        token: "{{ ansible_password }}"
      register: ise_creds


    - name: 02 - Get a List of the ISE Nodes
      cisco.ise.node_info:
        ise_hostname: "{{ ise_creds.data.data.ise_hostname }}"
        ise_username: "{{ ise_creds.data.data.ise_username }}"
        ise_password: "{{ ise_creds.data.data.ise_password }}"
        ise_verify: true
        ise_debug: false
      register: ise_node_list
      timeout: 120
```

This task only runs to query \ obtain a cred from Vault.

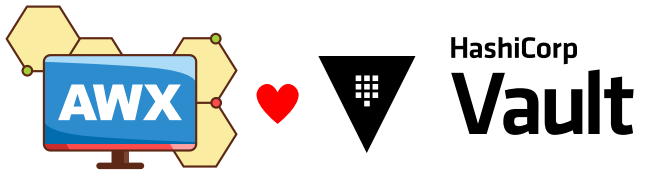
Not very efficient.

What if we need multiple creds?

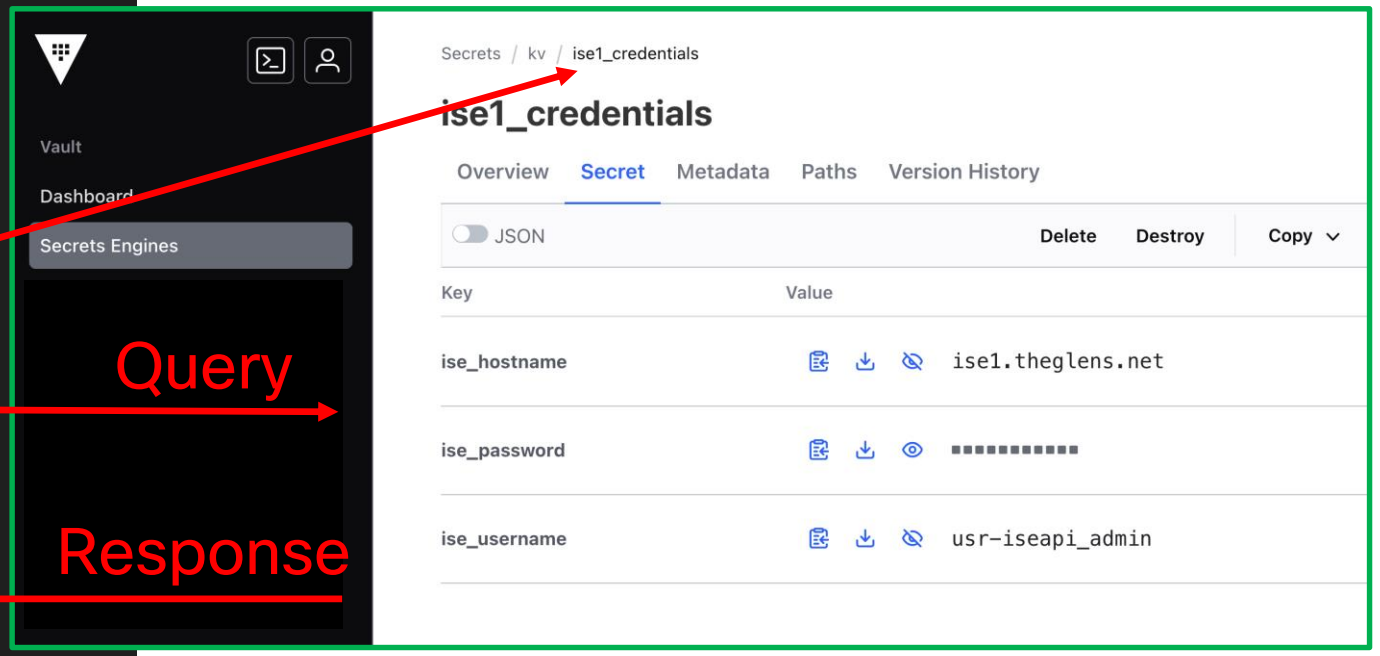
Copy \ paste this code into **how many** playbooks ?

How readable is this for the next person?

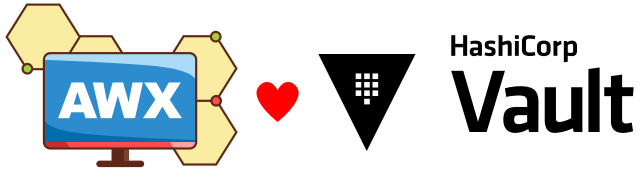
Lessons Learned
Limitation of Built-in Credentials




```yaml
---
- name: Get a List of the ISE Nodes v1.0
  hosts: localhost
  gather_facts: false

  tasks:

    - name: 01 - Read a ISE Credentials from Vault
      community.hashi_vault.vault_kv2_get:
        path: "ise1_credentials"
        url: "https://vault.theglens.net:8200"
        engine_mount_point: "kv"
        auth_method: token
        token: "{{ ansible_password }}"
      register: ise_creds


    - name: 02 - Get a List of the ISE Nodes
      cisco.ise.node_info:
        ise_hostname: "{{ ise_creds.data.data.ise_hostname }}"
        ise_username: "{{ ise_creds.data.data.ise_username }}"
        ise_password: "{{ ise_creds.data.data.ise_password }}"
        ise_verify: true
        ise_debug: false
      register: ise_node_list
      timeout: 120
```



Lessons Learned
Limitation of Built-in Credentials




```yaml
---
- name: Get a List of the ISE Nodes v1.0
  hosts: localhost
  gather_facts: false

  tasks:

    - name: 01 - Read a ISE Credentials from Vault
      community.hashi_vault.vault_kv2_get:
        path: "ise1_credentials"
        url: "https://vault.theglens.net:8200"
        engine_mount_point: "kv"
        auth_method: token
        token: "{{ ansible_password }}"
      register: ise_creds


    - name: 02 - Get a List of the ISE Nodes
      cisco.ise.node_info:
        ise_hostname: "{{ ise_creds.data.data.ise_hostname }}"
        ise_username: "{{ ise_creds.data.data.ise_username }}"
        ise_password: "{{ ise_creds.data.data.ise_password }}"
        ise_verify: true
        ise_debug: false
      register: ise_node_list
      timeout: 120
```

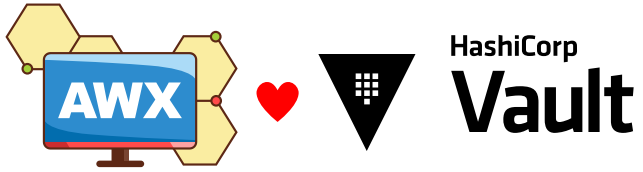
Limitation
AWX Machine Credential
ansible_password

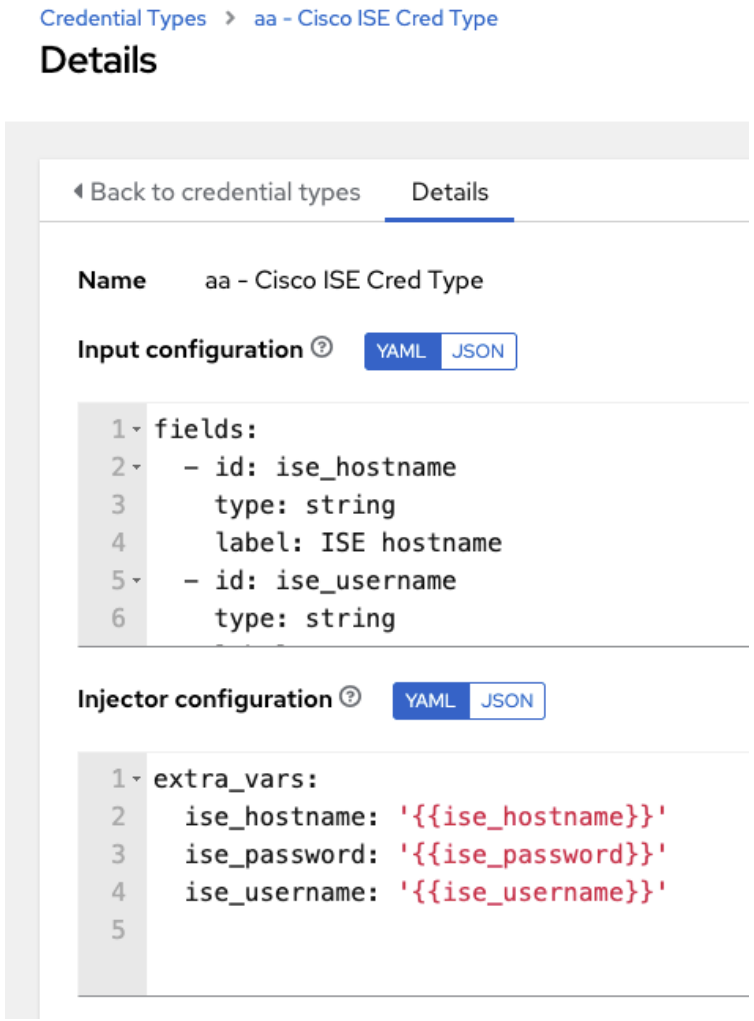
- Reserved variable name
- Limited usage
- Not designed to be used this way
- Doesn't feel good, it's a hack

Why did I do it this way?

- Familiar with ‘machine credential’ / SSH cred in Ansible
  - ansible_username / ansible_passord
- Knew that AWX would encrypt ansible_password
- Comfortable storing api_key, tokens, in ansible_password
- Knew it was a hack but it worked, till it didn’t

Welcome, Custom Credential Type





Input configuration

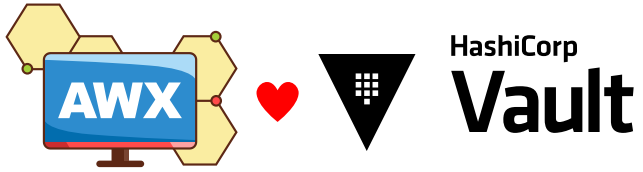
```
fields:
  - id: ise_hostname
    type: string
    label: ISE hostname
  - id: ise_username
    type: string
    label: Username
  - id: ise_password
    type: string
    label: Password
    secret: true
required:
  - ise_hostname
  - ise_username
  - ise_password
```

Injector configuration

```
extra_vars:
  ise_hostname: '{{ise_hostname}}'
  ise_password: '{{ise_password}}'
  ise_username: '{{ise_username}}'
```

https://docs.ansible.com/automation-controller/4.4/html/userguide/credential_types.html

Welcome, Custom Credential Type



Credential Types > aa - Cisco ISE Cred Type

Details

Back to credential types | Details

Name aa - Cisco ISE Cred Type

Input configuration (YAML | JSON)

```
1 fields:
2   - id: ise_hostname
3     type: string
4     label: ISE hostname
5   - id: ise_username
6     type: string
```

Injector configuration (YAML | JSON)

```
1 extra_vars:
2   ise_hostname: '{{ise_hostname}}'
3   ise_password: '{{ise_password}}'
4   ise_username: '{{ise_username}}'
5
```

Input configuration

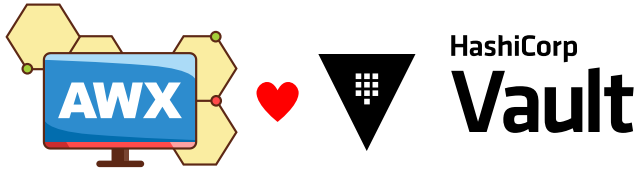
```
fields:
  - id: ise_hostname
    type: string
    label: ISE hostname
  - id: ise_username
    type: string
    label: Username
  - id: ise_password
    type: string
    label: Password
    secret: true
required:
  - ise_hostname
  - ise_username
  - ise_password
```

Injector configuration

```
extra_vars:
  ise_hostname: '{{ise_hostname}}'
  ise_password: '{{ise_password}}'
  ise_username: '{{ise_username}}'
```

https://docs.ansible.com/automation-controller/4.4/html/userguide/credential_types.html

Welcome, Custom Credential Type



Credential Types > aa - Cisco ISE Cred Type

Details

Back to credential types | Details

Name aa - Cisco ISE Cred Type

Input configuration (YAML | JSON)

```yaml
fields:
  - id: ise_hostname
    type: string
    label: ISE hostname
  - id: ise_username
    type: string
```

Injector configuration (YAML | JSON)

```yaml
extra_vars:
  ise_hostname: '{{ise_hostname}}'
  ise_password: '{{ise_password}}'
  ise_username: '{{ise_username}}'
```

Input configuration

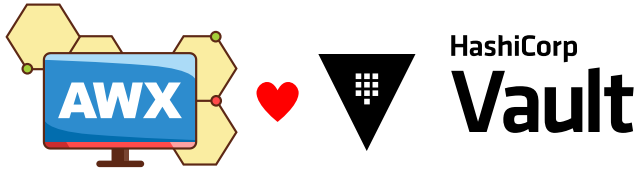
```yaml
fields:
  - id: ise_hostname
    type: string
    label: ISE hostname
  - id: ise_username
    type: string
    label: Username
  - id: ise_password
    type: string
    label: Password
    secret: true
required:
  - ise_hostname
  - ise_username
  - ise_password
```

Injector configuration

```yaml
extra_vars:
  ise_hostname: '{{ise_hostname}}'
  ise_password: '{{ise_password}}'
  ise_username: '{{ise_username}}'
```

https://docs.ansible.com/automation-controller/4.4/html/userguide/credential_types.html

Welcome, Custom Credential Type



Credential Types > aa - Cisco ISE Cred Type

Details

Back to credential types | Details

Name aa - Cisco ISE Cred Type

Input configuration YAML JSON

```yaml
fields:
  - id: ise_hostname
    type: string
    label: ISE hostname
  - id: ise_username
    type: string
```

Injector configuration YAML JSON

```yaml
extra_vars:
  ise_hostname: '{{ise_hostname}}'
  ise_password: '{{ise_password}}'
  ise_username: '{{ise_username}}'
```

Input configuration

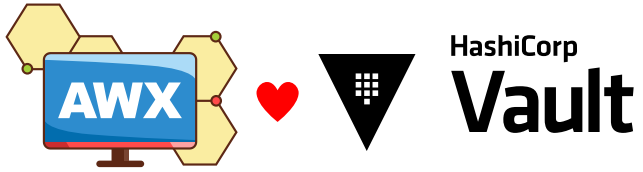
```yaml
fields:
  - id: ise_hostname
    type: string
    label: ISE hostname
  - id: ise_username
    type: string
    label: Username
  - id: ise_password
    type: string
    label: Password
    secret: true
required:
  - ise_hostname
  - ise_username
  - ise_password
```

Injector configuration

```yaml
extra_vars:
  ise_hostname: '{{ise_hostname}}'
  ise_password: '{{ise_password}}'
  ise_username: '{{ise_username}}'
```

https://docs.ansible.com/automation-controller/4.4/html/userguide/credential_types.html

Welcome, Custom Credential Type



Credential Types > aa - Cisco ISE Cred Type

Details

Back to credential types | Details

Name aa - Cisco ISE Cred Type

Input configuration (YAML | JSON)

```yaml
fields:
  - id: ise_hostname
    type: string
    label: ISE hostname
  - id: ise_username
    type: string
```

Injector configuration (YAML | JSON)

```yaml
extra_vars:
  ise_hostname: '{{ise_hostname}}'
  ise_password: '{{ise_password}}'
  ise_username: '{{ise_username}}'
```

Input configuration

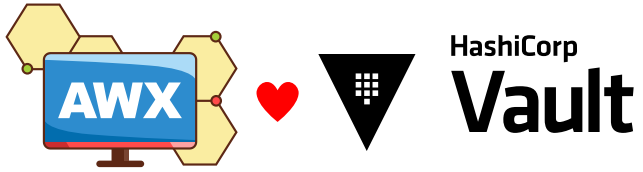
```yaml
fields:
  - id: ise_hostname
    type: string
    label: ISE hostname
  - id: ise_username
    type: string
    label: Username
  - id: ise_password
    type: string
    label: Password
    secret: true
required:
  - ise_hostname
  - ise_username
  - ise_password
```

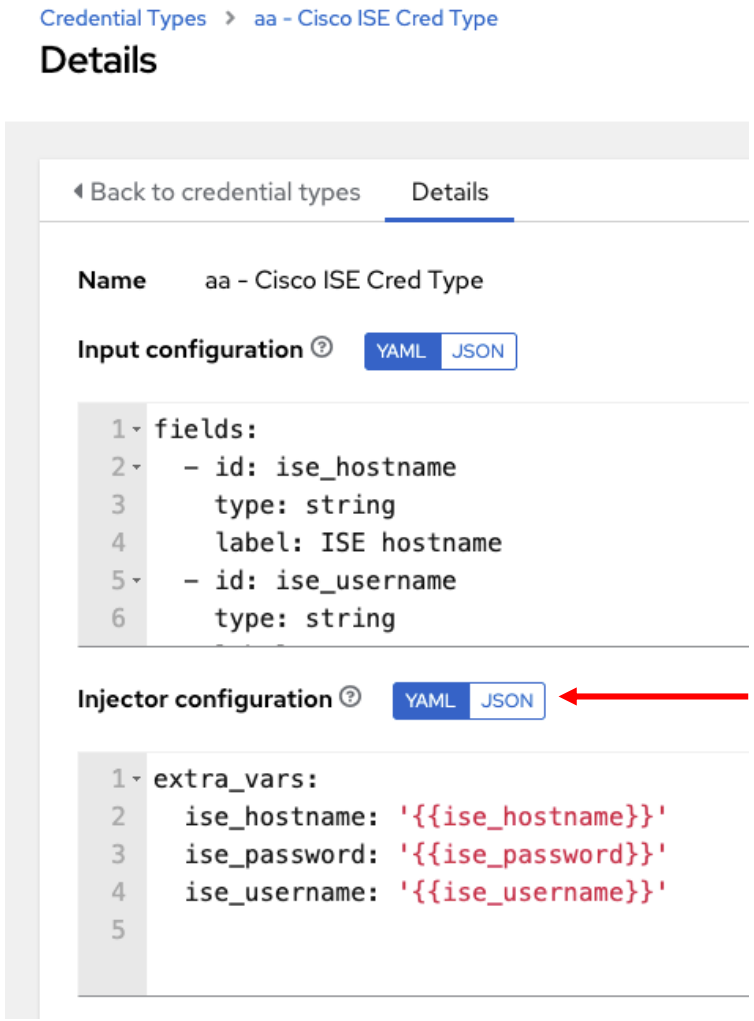
Injector configuration

```yaml
extra_vars:
  ise_hostname: '{{ise_hostname}}'
  ise_password: '{{ise_password}}'
  ise_username: '{{ise_username}}'
```

https://docs.ansible.com/automation-controller/4.4/html/userguide/credential_types.html

Welcome, Custom Credential Type





Input configuration

```
fields:
  - id: ise_hostname
    type: string
    label: ISE hostname
  - id: ise_username
    type: string
    label: Username
  - id: ise_password
    type: string
    label: Password
    secret: true
required:
  - ise_hostname
  - ise_username
  - ise_password
```

Injector configuration

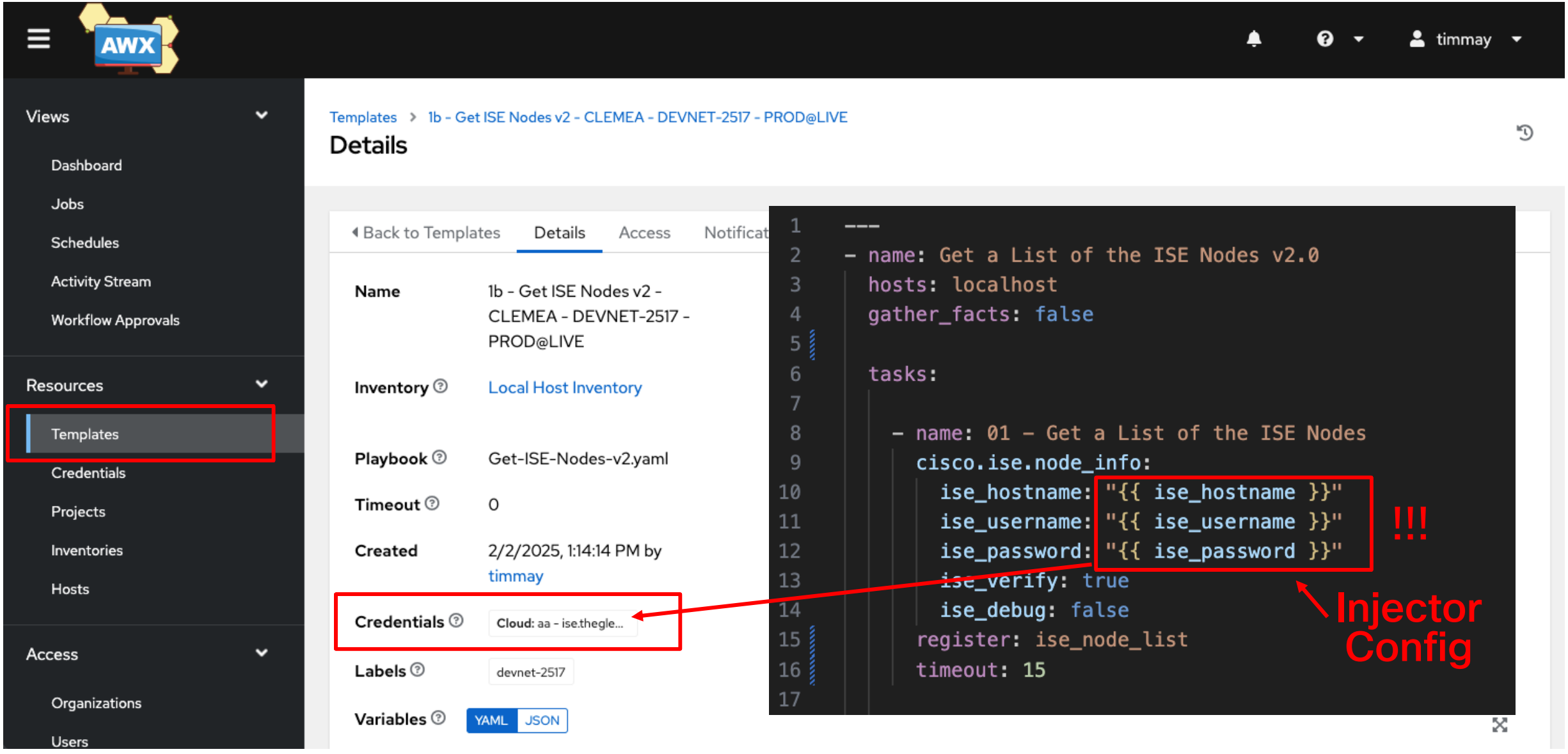
```
extra_vars:
  ise_hostname: '{{ise_hostname}}'
  ise_password: '{{ise_password}}'
  ise_username: '{{ise_username}}'
```

id

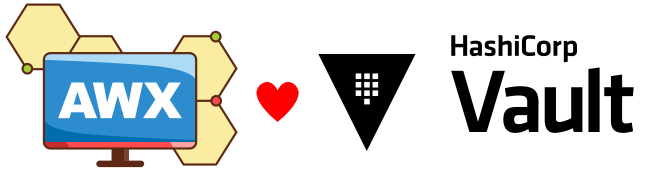
vars in playbook

https://docs.ansible.com/automation-controller/4.4/html/userguide/credential_types.html

Template to Playbook Cred Mapping



Old vs New Playbook




```yaml
---
- name: Get a List of the ISE Nodes v1.0
  hosts: localhost
  gather_facts: false

  tasks:

    - name: 01 - Read a ISE Credentials from Vault
      community.hashi_vault.vault_kv2_get:
        path: "ise1_credentials"
        url: "https://vault.theglens.net:8200"
        engine_mount_point: "kv"
        auth_method: token
        token: "{{ ansible_password }}"
      register: ise_creds


    - name: 02 - Get a List of the ISE Nodes
      cisco.ise.node_info:
        ise_hostname: "{{ ise_creds.data.data.ise_hostname }}"
        ise_username: "{{ ise_creds.data.data.ise_username }}"
        ise_password: "{{ ise_creds.data.data.ise_password }}"
        ise_verify: true
        ise_debug: false
      register: ise_node_list
      timeout: 120
```

Which credential should we use?

```yaml
---
- name: Get a List of the ISE Nodes v2.0
  hosts: localhost
  gather_facts: false

  tasks:

    - name: 01 - Get a List of the ISE Nodes
      cisco.ise.node_info:
        ise_hostname: "{{ ise_hostname }}"
        ise_username: "{{ ise_username }}"
        ise_password: "{{ ise_password }}"
        ise_verify: true
        ise_debug: false
      register: ise_node_list
      timeout: 15

```

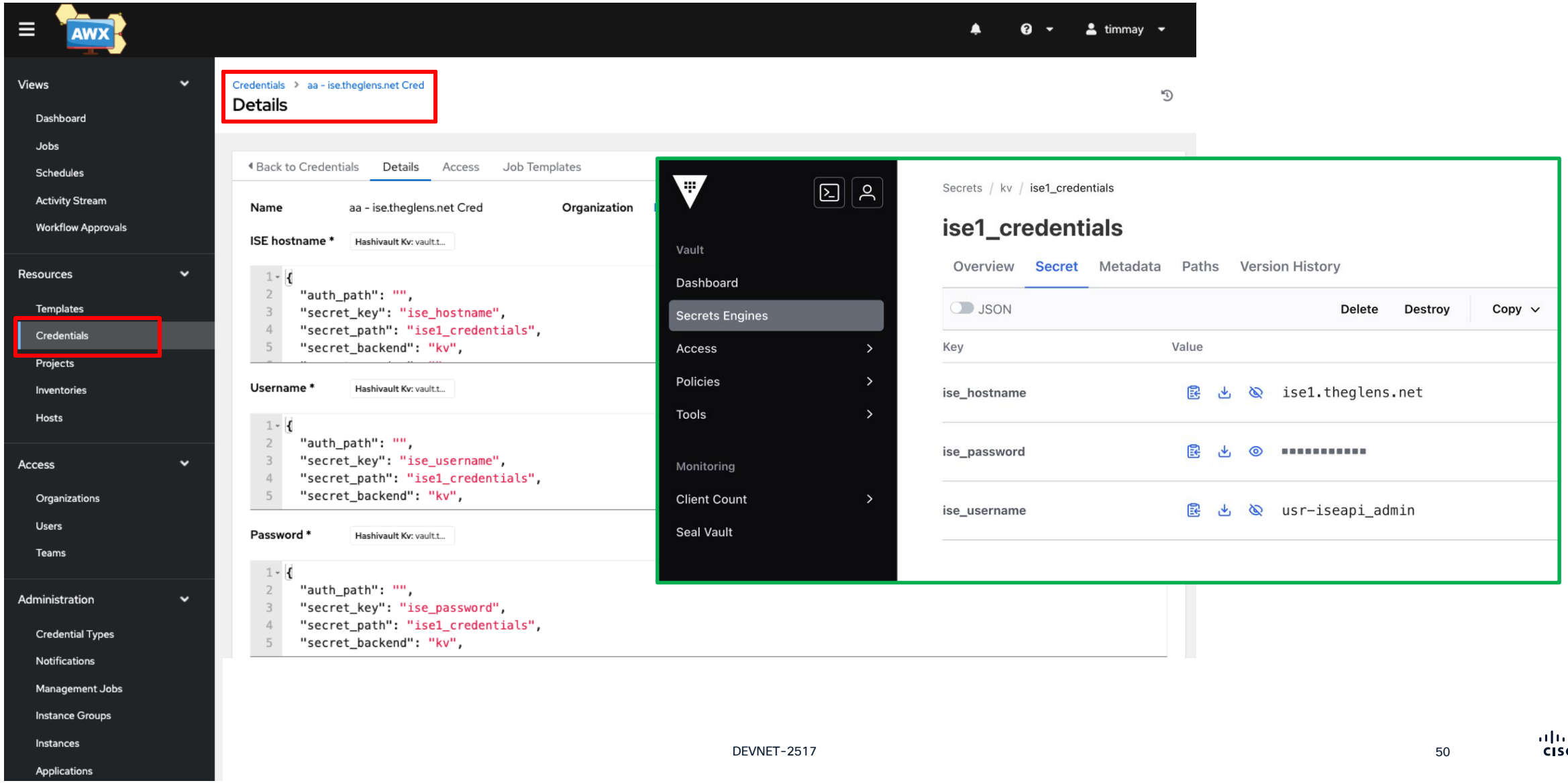
Easy & Reusable

DEMO – Build a Simple ISE Cred

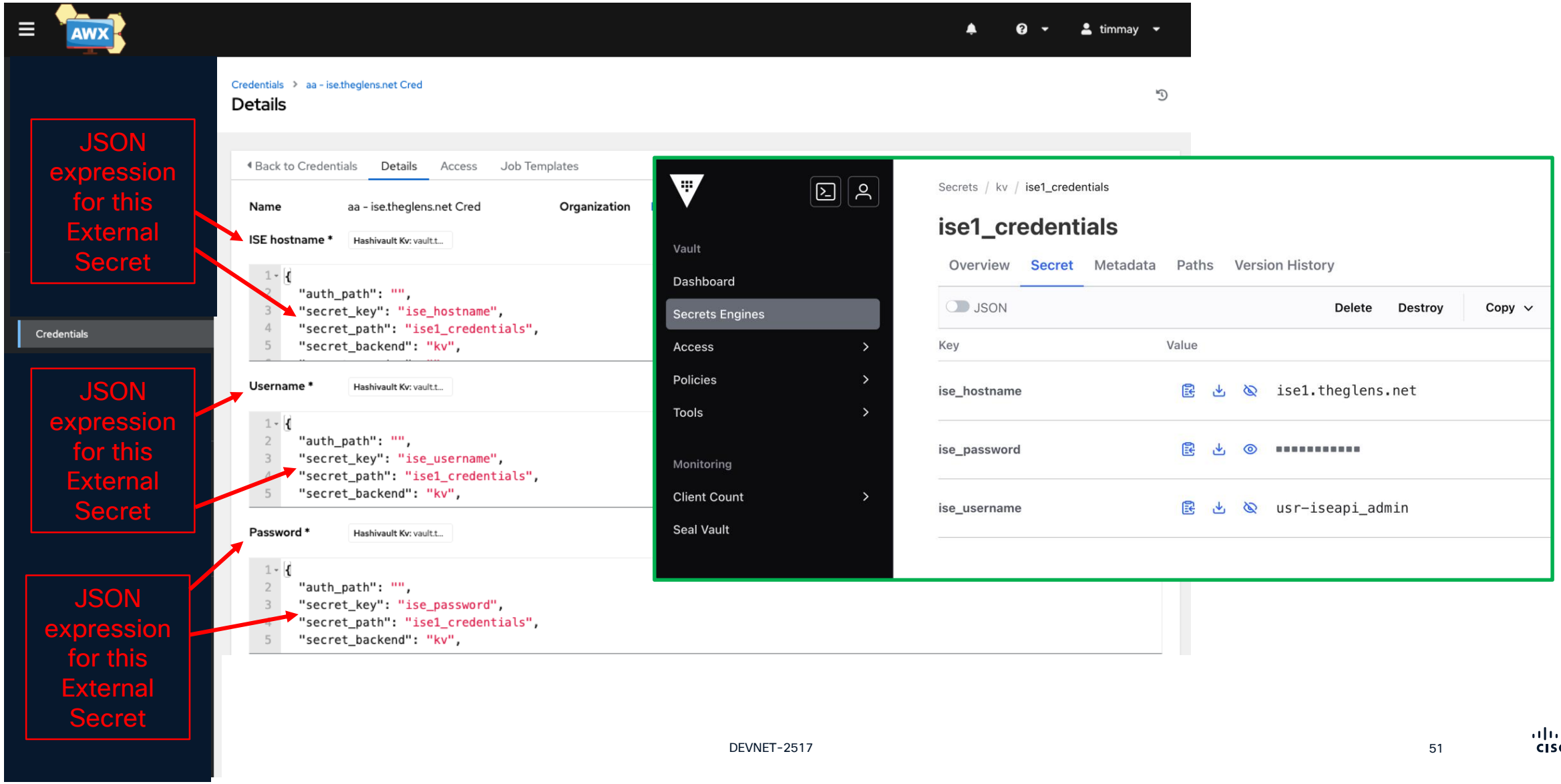
- Show Existing Custom Credential Type – aa – Cisco ISE Cred Type

- Create new Credential
- Type aa– ISE
- Name

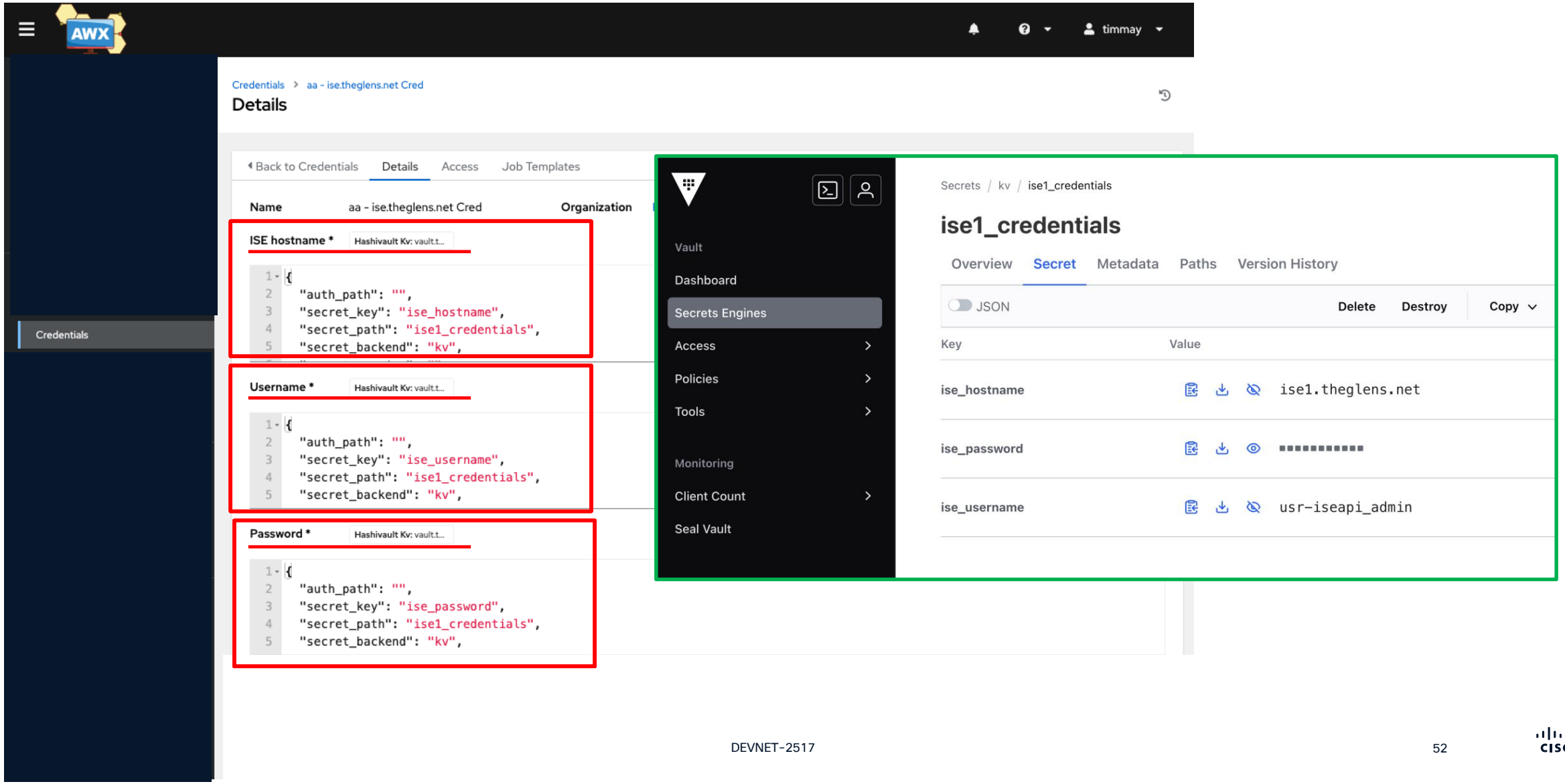
Custom Cred to Vault Mapping



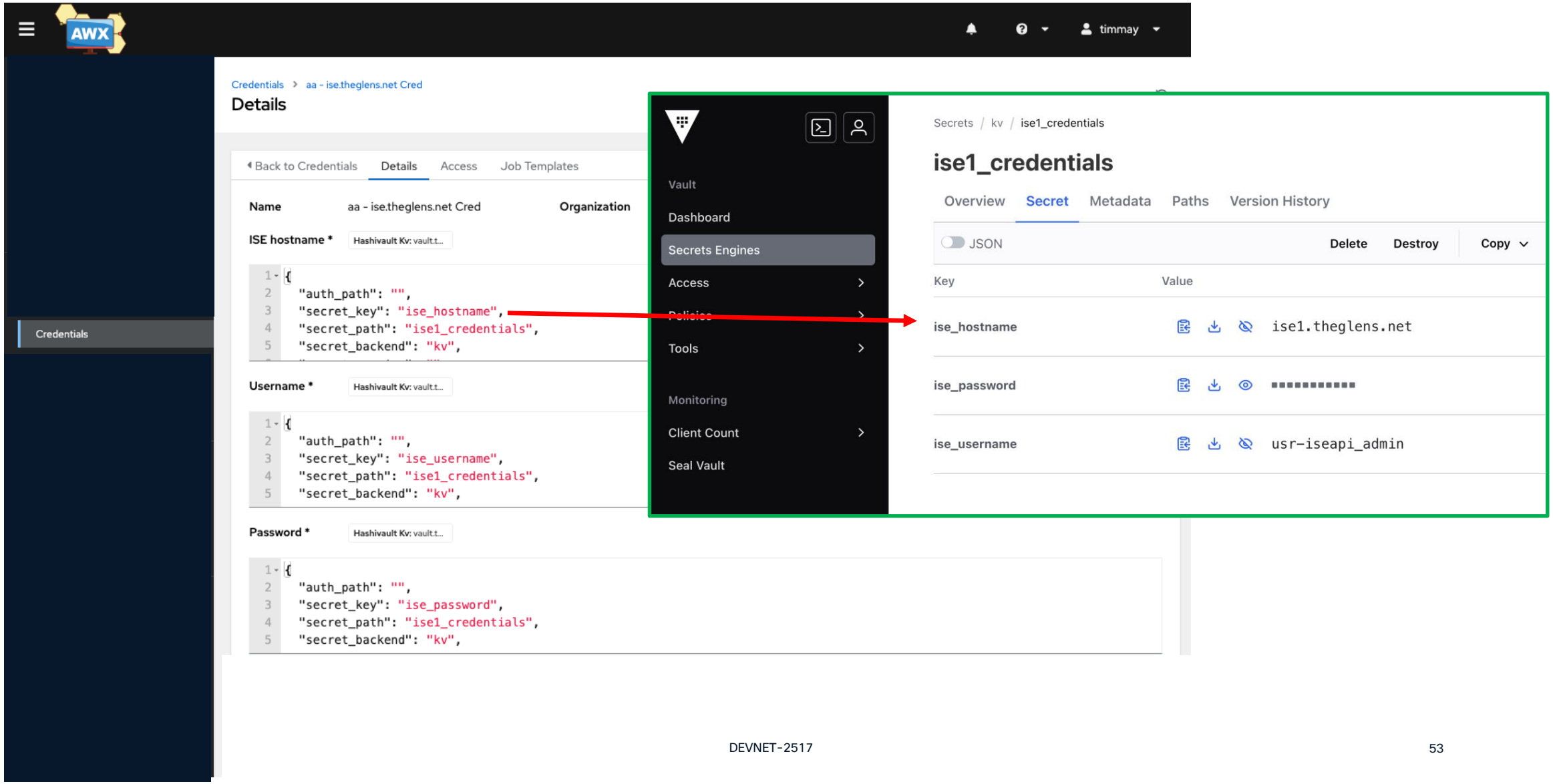
Custom Cred to Vault Mapping



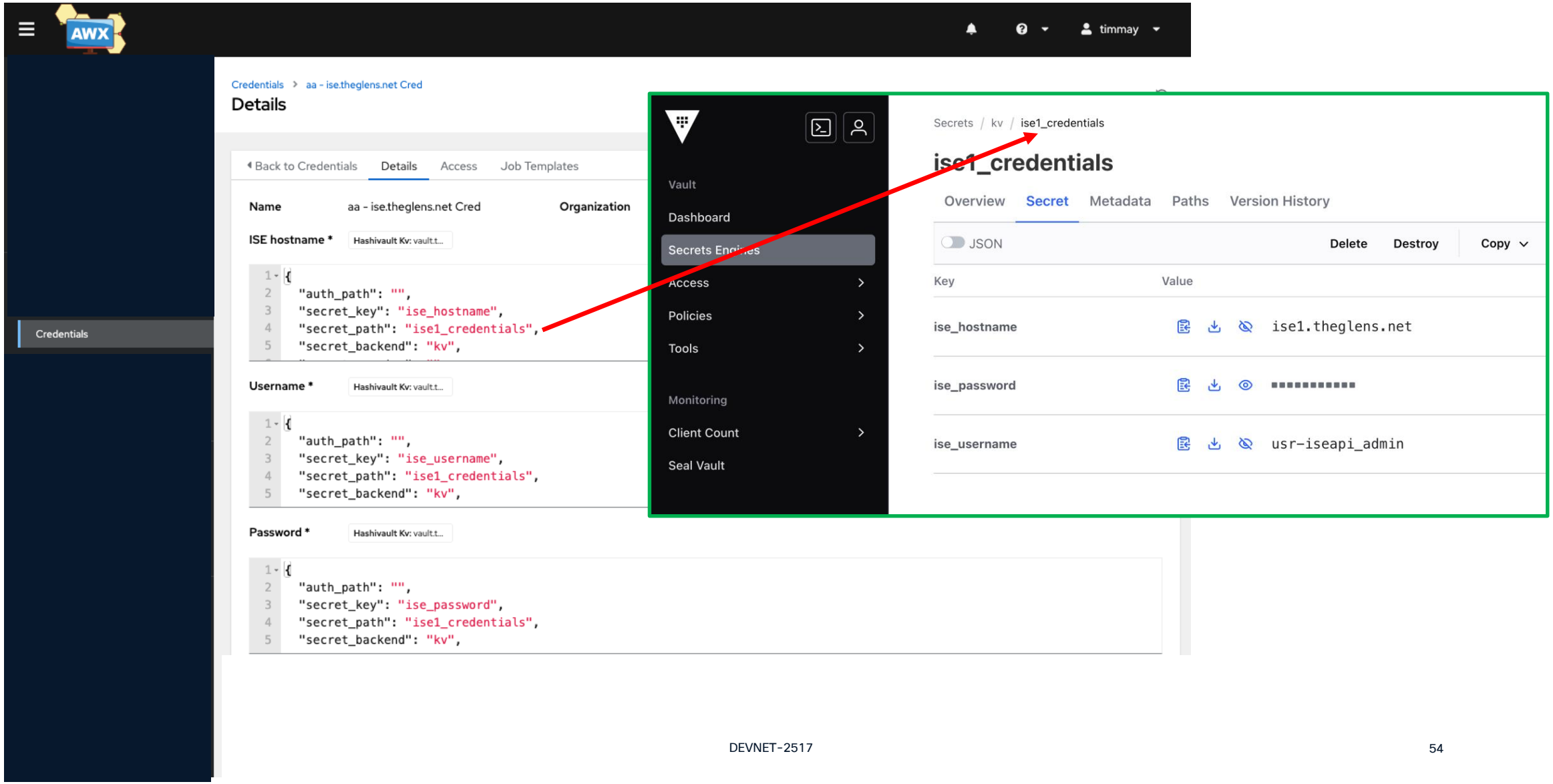
Custom Cred to Vault Mapping



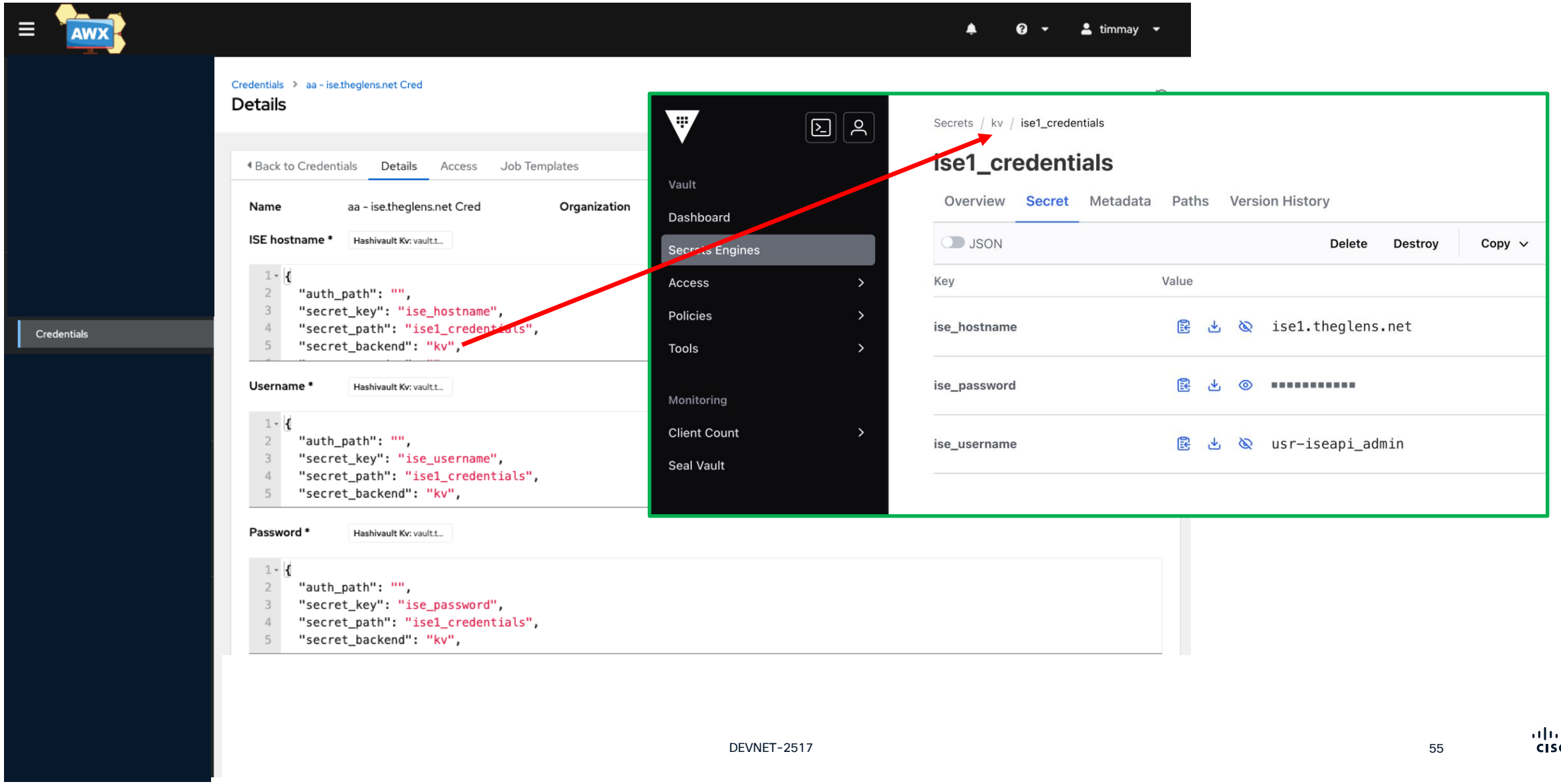
Custom Cred to Vault Mapping



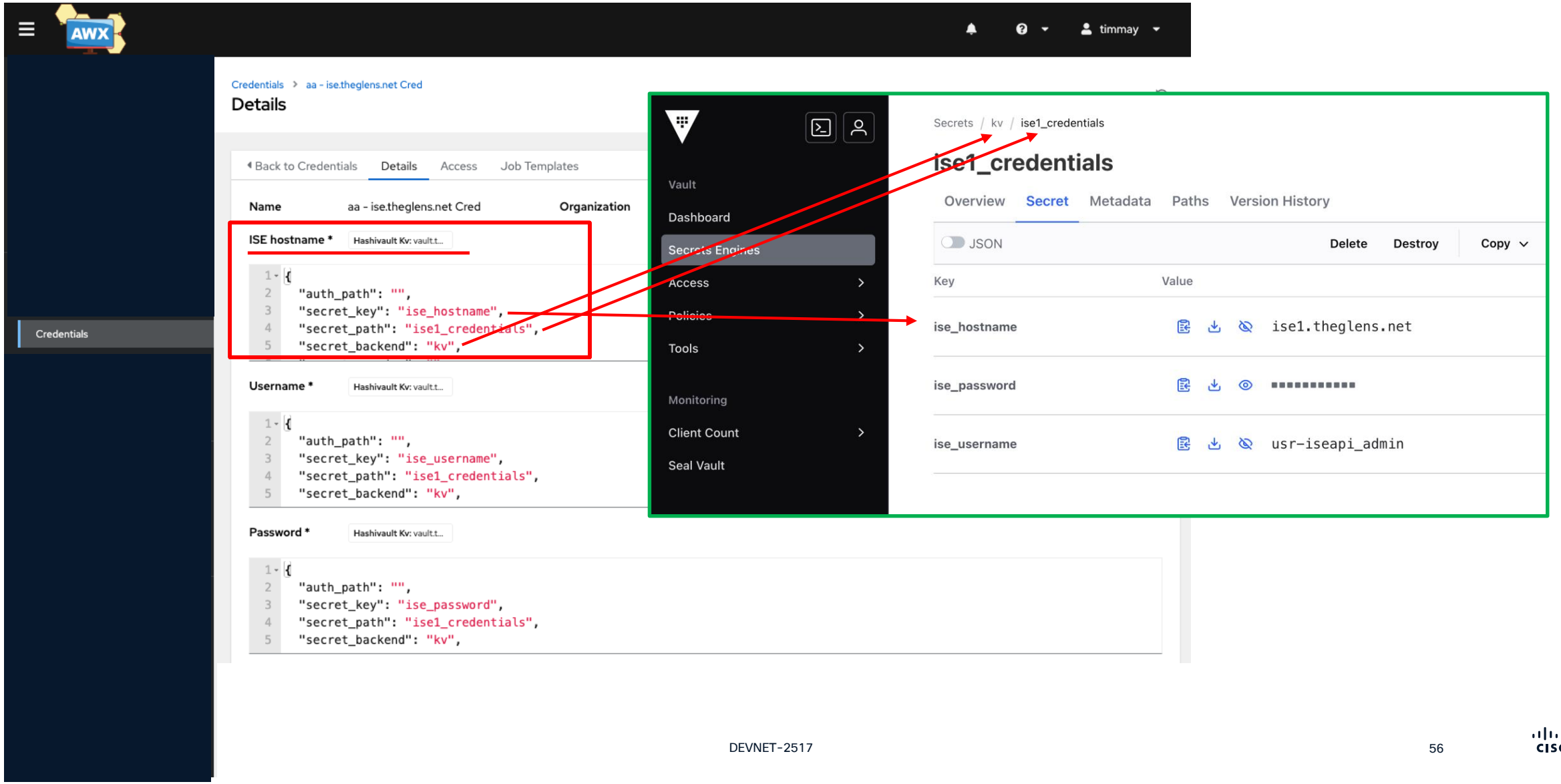
Custom Cred to Vault Mapping



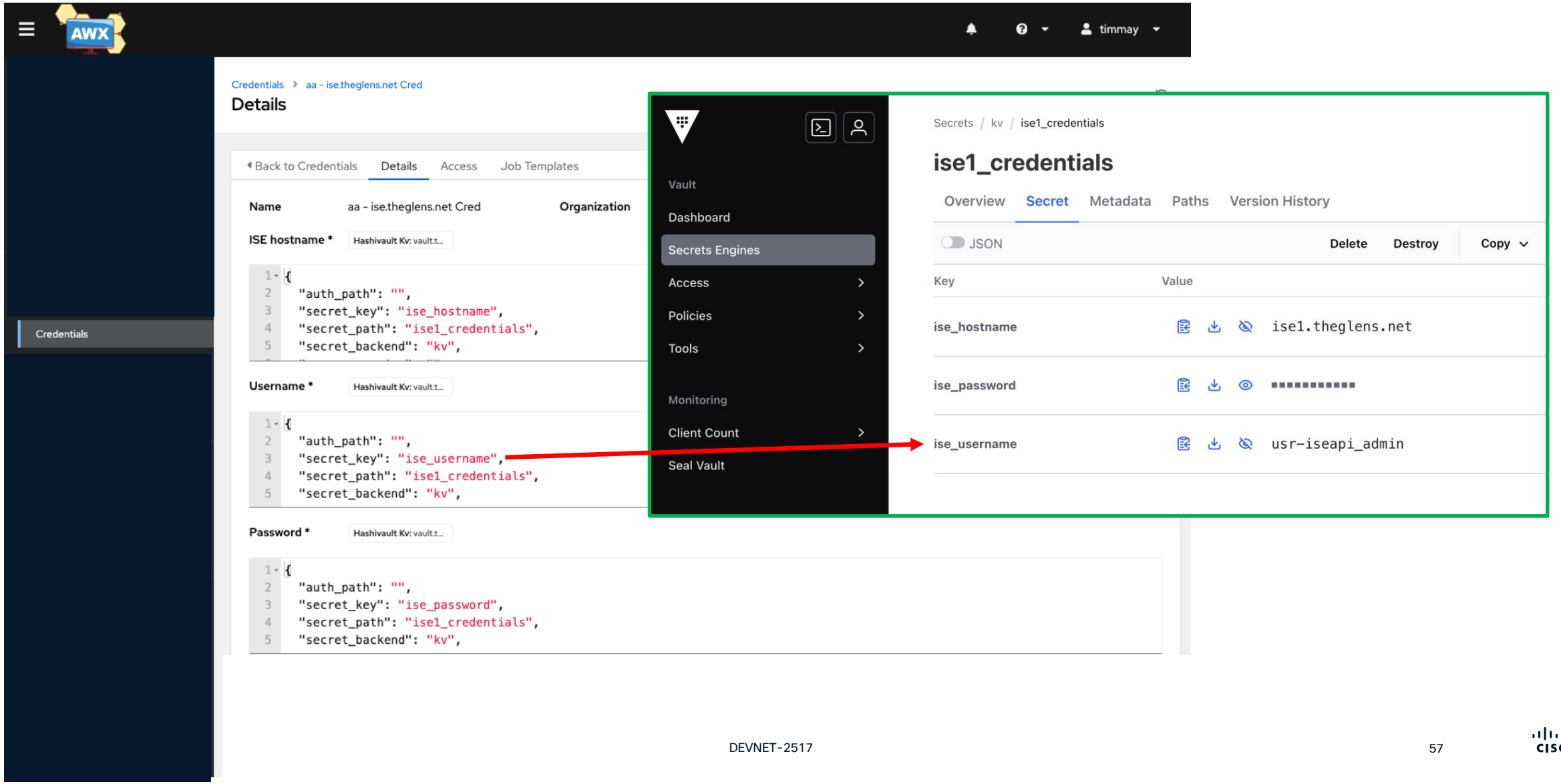
Custom Cred to Vault Mapping



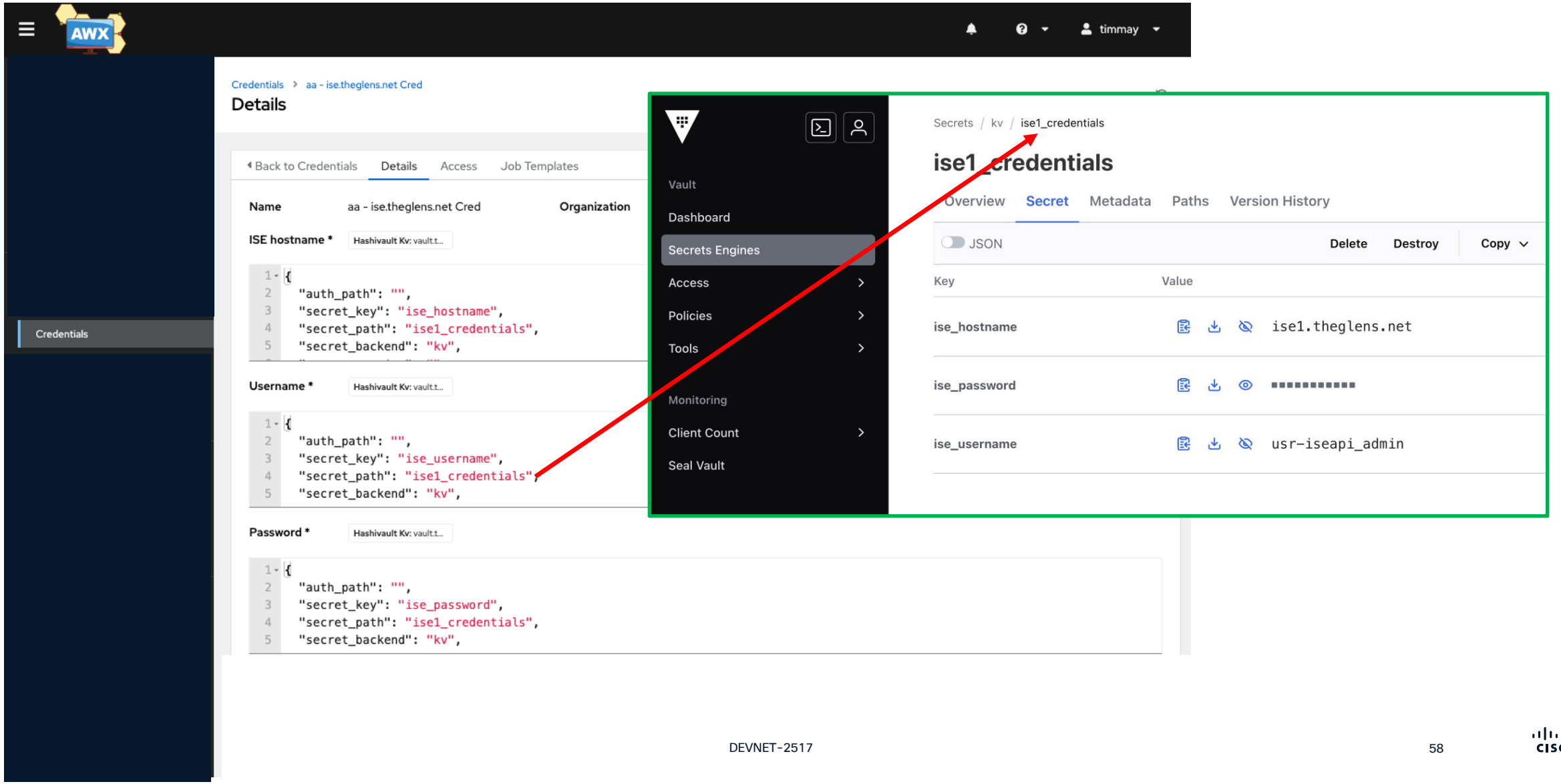
Custom Cred to Vault Mapping



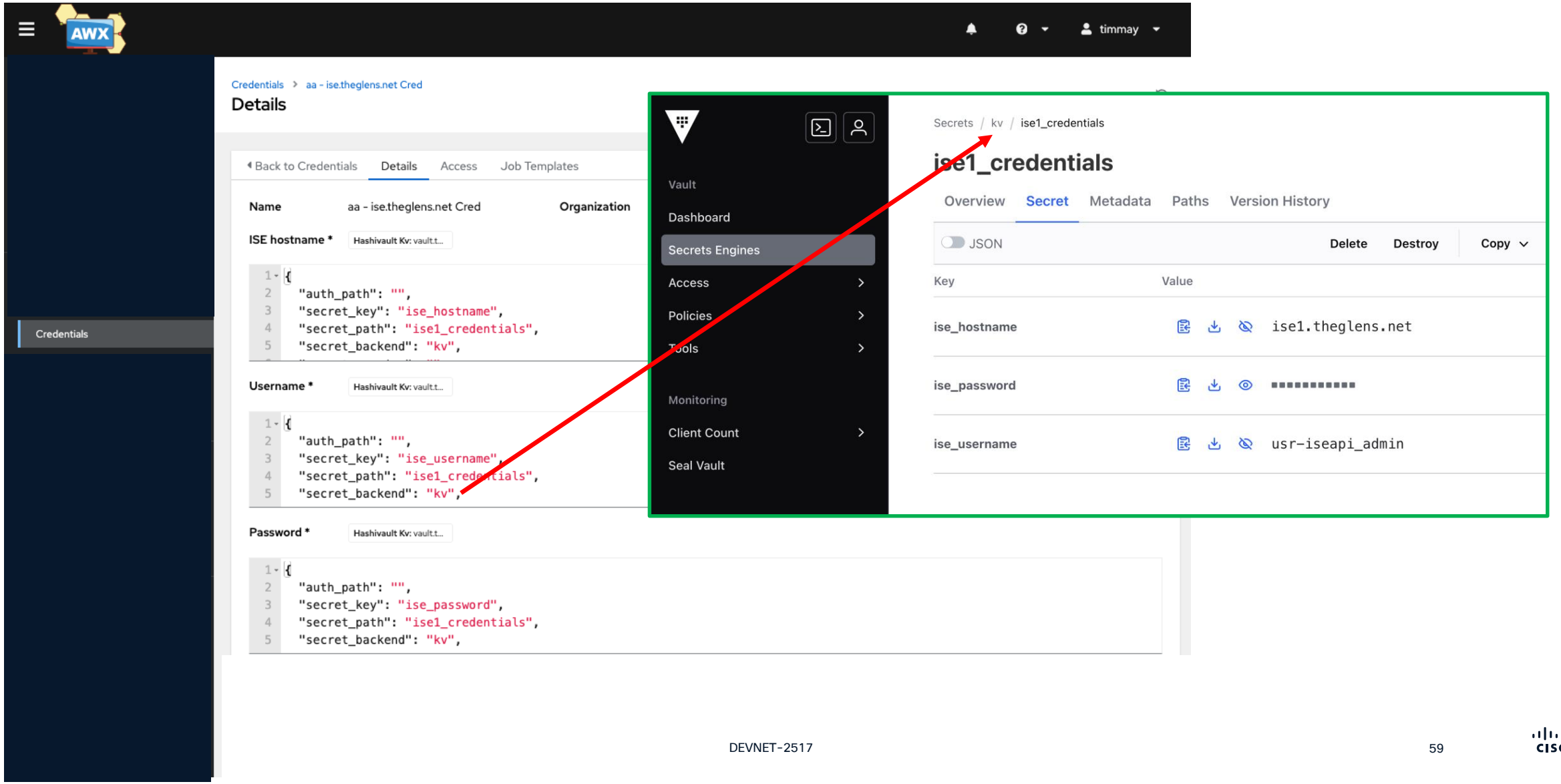
Custom Cred to Vault Mapping



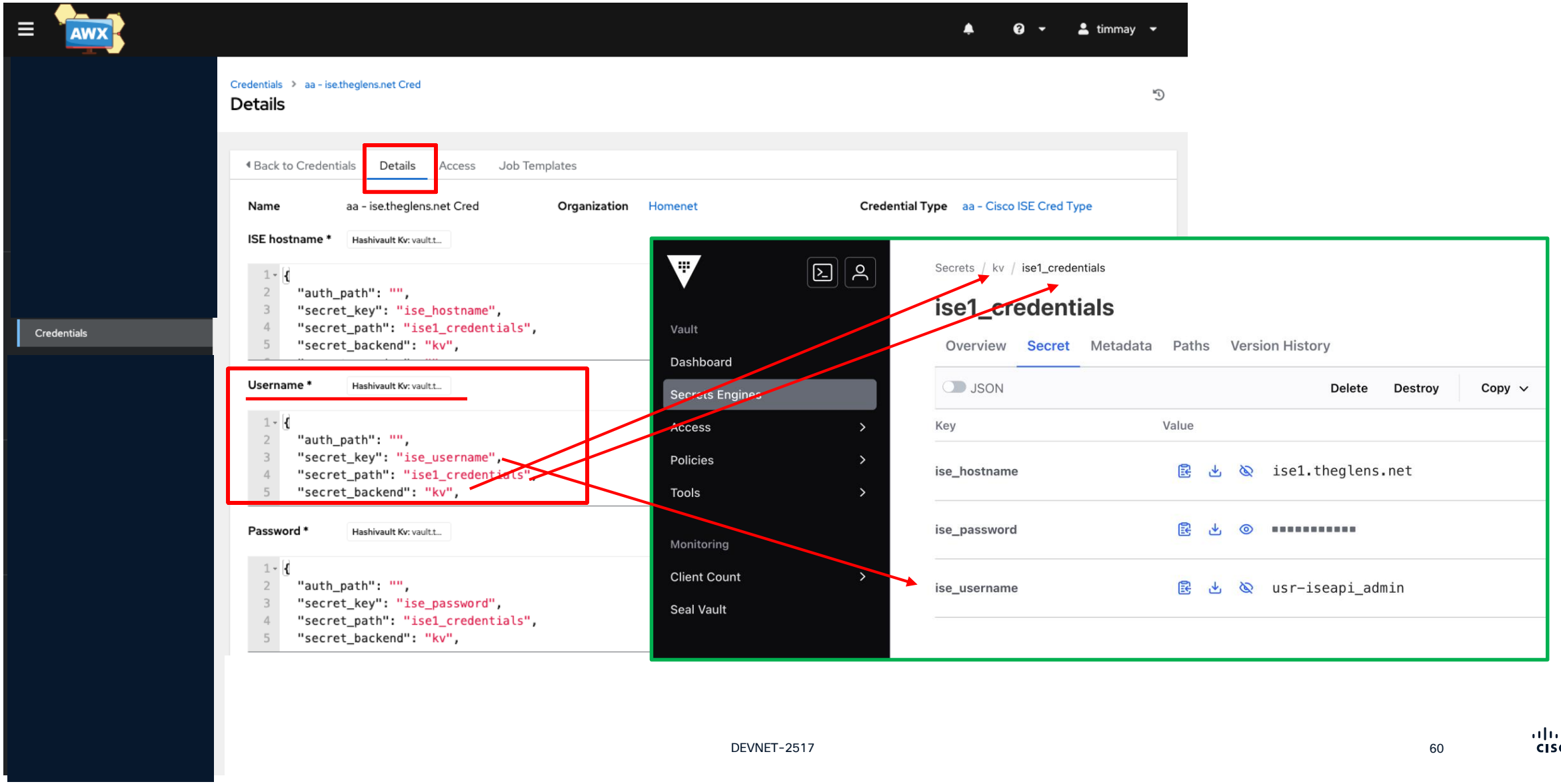
Custom Cred to Vault Mapping



Custom Cred to Vault Mapping

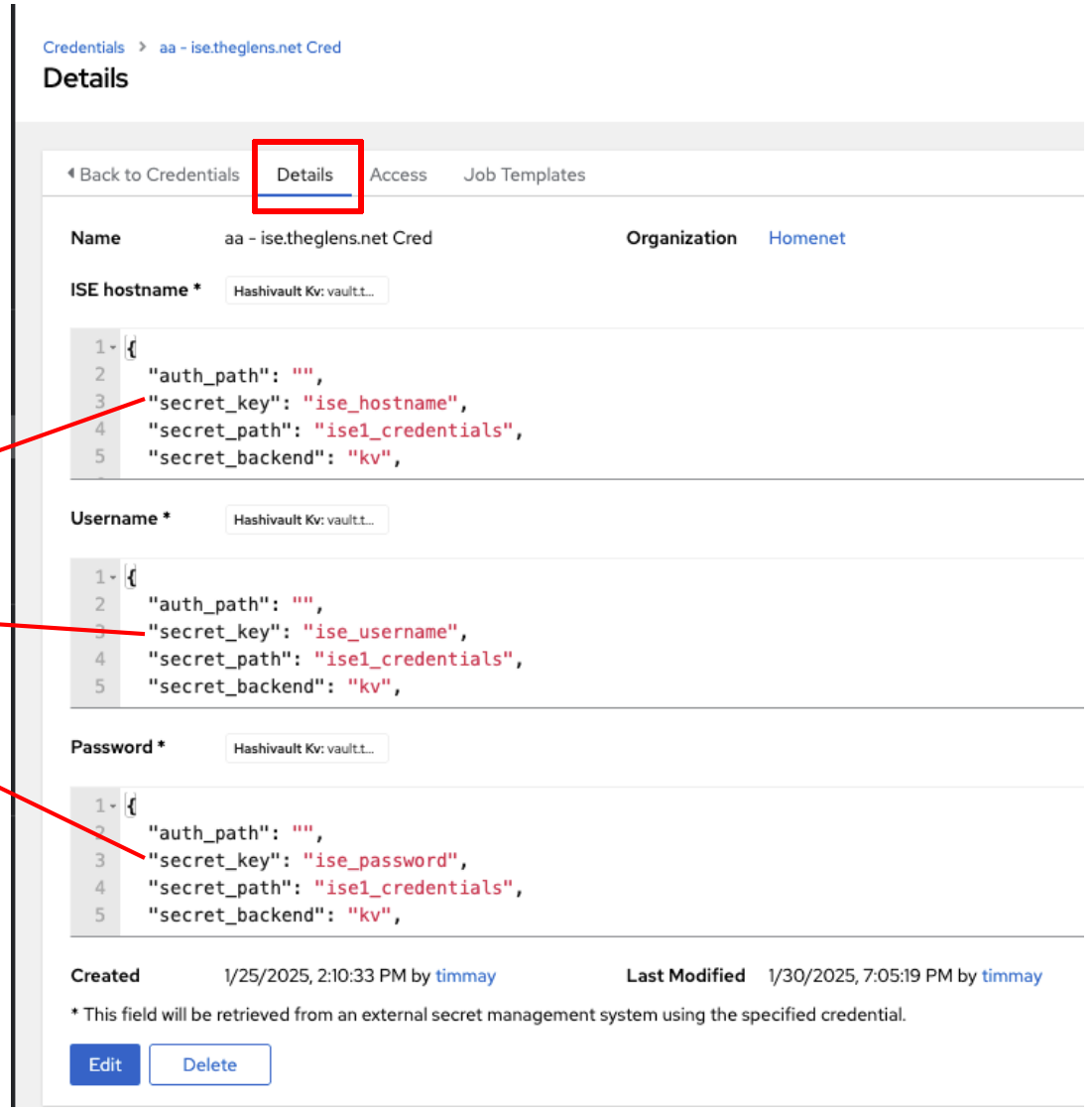


Custom Cred to Vault Mapping



Playbook to Custom Cred Mapping

```yaml
---
- name: Get a List of the ISE Nodes v2.0
  hosts: localhost
  gather_facts: false

  tasks:

    - name: 01 - Get a List of the ISE Nodes
      cisco.ise.node_info:
        ise_hostname: "{{ ise_hostname }}"
        ise_username: "{{ ise_username }}"
        ise_password: "{{ ise_password }}"
        ise_verify: true
        ise_debug: false
      register: ise_node_list
      timeout: 15

```

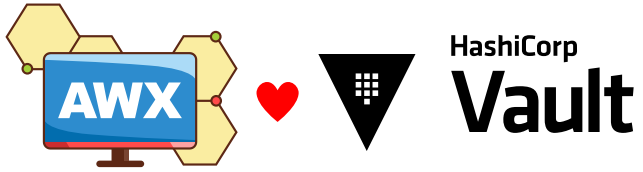


Sorry for making you focus on all those arrows.

There will be a Cisco certification test after this session, and you will have to drag and drop from one side to the other.

I hope you were paying attention!!!

Custom Cred Reuse



Credentials › aa - ise.theglens.net Cred

Details

Back to Credentials | Details | Access | Job Templates

Name aa - ise.theglens.net Cred **Organization** Homenet

ISE hostname * Hashivault Kv: vault.t...

```
{
  "auth_path": "",
  "secret_key": "ise_hostname",
  "secret_path": "ise1_credentials",
  "secret_backend": "kv",
```

Username * Hashivault Kv: vault.t...

```
{
  "auth_path": "",
  "secret_key": "ise_username",
  "secret_path": "ise1_credentials",
  "secret_backend": "kv",
```

Password * Hashivault Kv: vault.t...

```
{
  "auth_path": "",
  "secret_key": "ise_password",
  "secret_path": "ise1_credentials",
  "secret_backend": "kv",
```

Created 1/25/2025, 2:10:33 PM by timmay **Last Modified** 1/30/2025, 7:05:19 PM by timmay

* This field will be retrieved from an external secret management system using the specified credential.

Edit | Delete

Credentials › aa - ise.theglens.net Cred

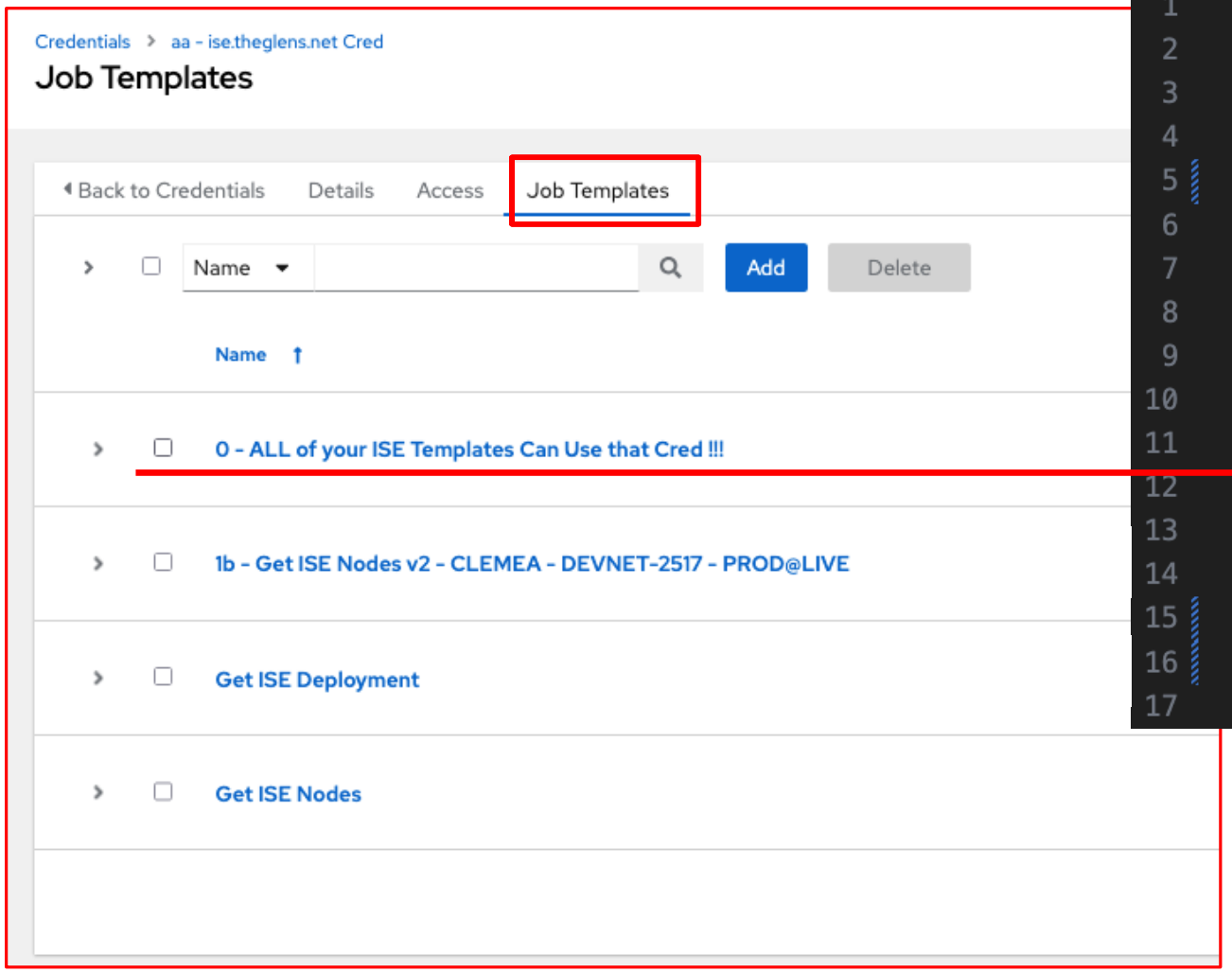
Job Templates

Back to Credentials | Details | Access | Job Templates

Name | Add | Delete

Name	Type
0 - ALL of your ISE Templates Can Use that Cred !!!	Job Template
1b - Get ISE Nodes v2 - PROD@LIVE	Job Template
Get ISE Deployment	Job Template
Get ISE Nodes	Job Template

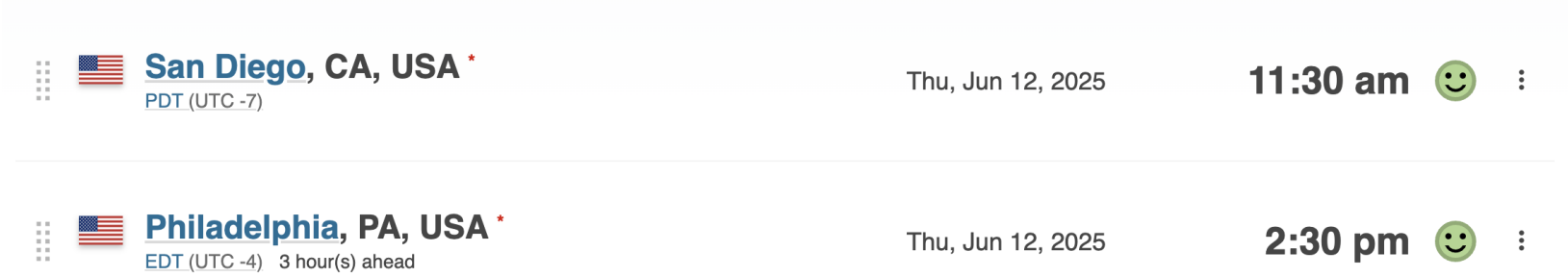
Custom Cred Reuse




```yaml
---
- name: Get a List of the ISE Nodes v2.0
  hosts: localhost
  gather_facts: false

  tasks:

    - name: 01 - Get a List of the ISE Nodes
      cisco.ise.node_info:
        ise_hostname: "{{ ise_hostname }}"
        ise_username: "{{ ise_username }}"
        ise_password: "{{ ise_password }}"
        ise_verify: true
        ise_debug: false
      register: ise_node_list
      timeout: 15

```

Demo – External Secret Management

- Replace the Certificate on ISE during the Business Day

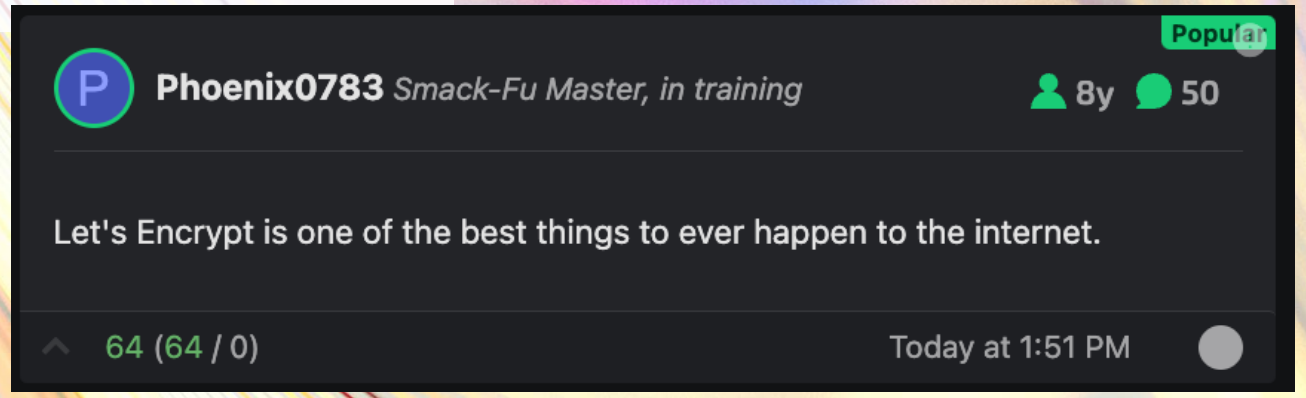


- Show Playbooks
- ISE Should be finished restarting soon

Let's Encrypt

Demo

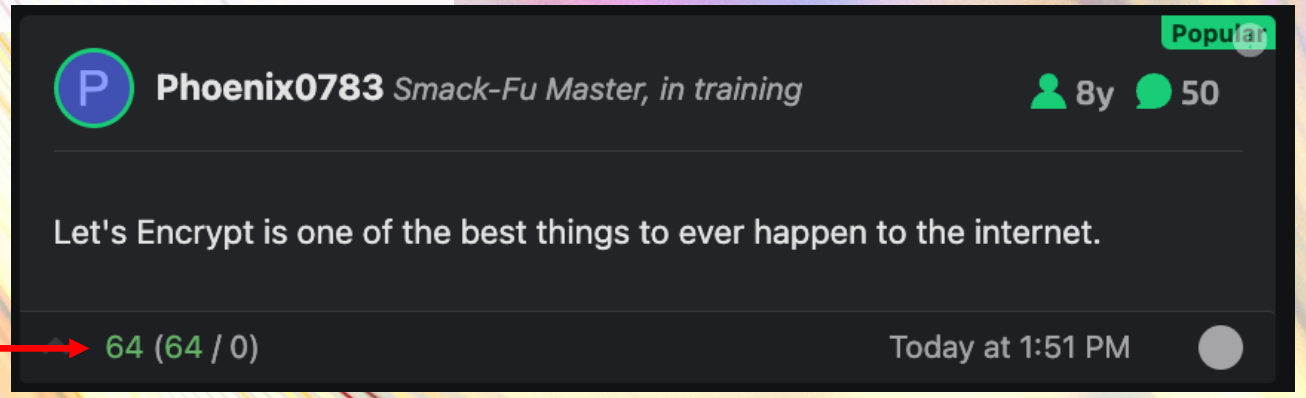
Certificate Automation



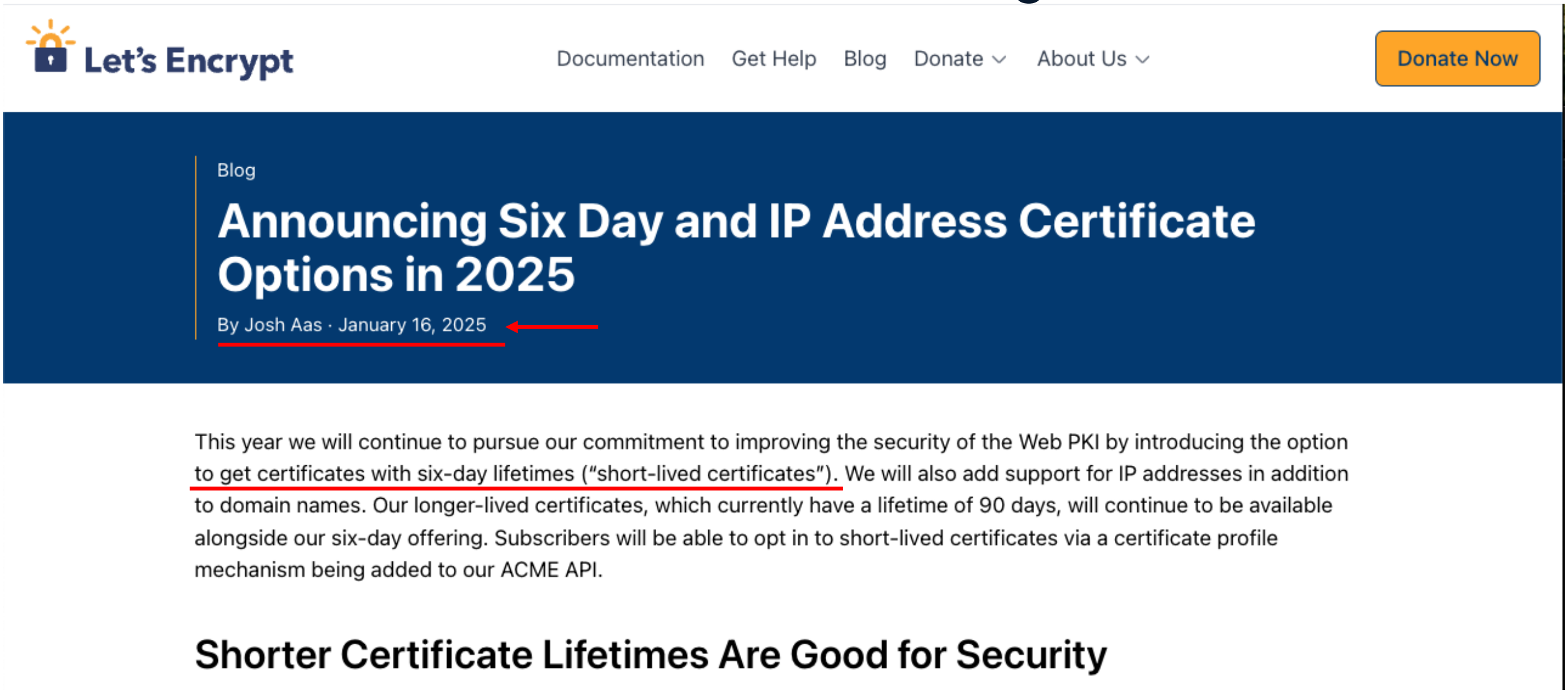
Demo

Certificate Automation

No
Downvotes

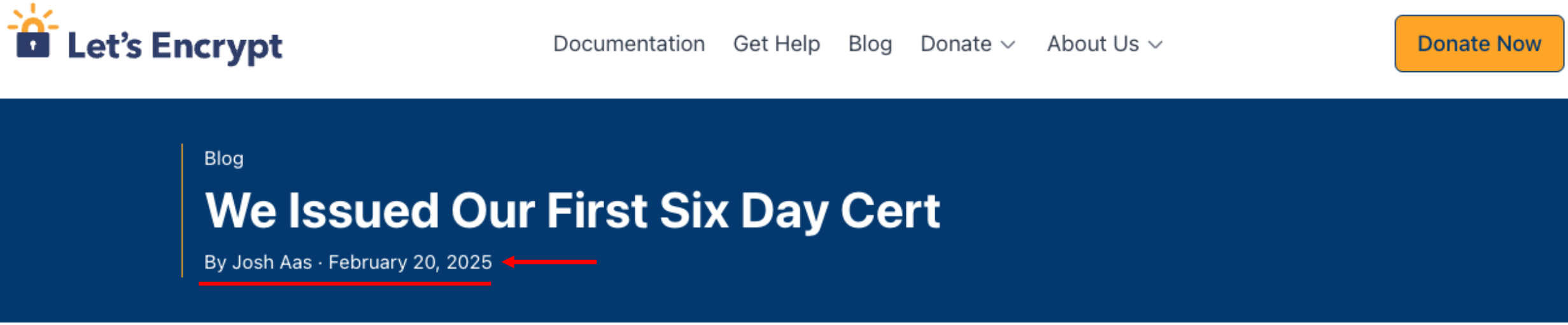


Shorter Certificate Lifetimes are Coming...



https://letsencrypt.org/2025/01/16/6-day-and-ip-certs/

Shorter Certificate Lifetimes are Coming...



Earlier this year we announced our intention to introduce short-lived certificates with lifetimes of six days as an option for our subscribers. Yesterday we issued our first short-lived certificate. You can see the certificate at the bottom of our post, or here thanks to Certificate Transparency logs. We issued it to ourselves and then immediately revoked it so we can observe the certificate's whole lifecycle. This is the first step towards making short-lived certificates available to all subscribers.

The next step is for us to make short-lived certificates available to a small set of our subscribers so we can make sure our systems scale as expected prior to general availability. We expect this next phase to begin during Q2 of this year.

We expect short-lived certificates to be generally available by the end of this year.

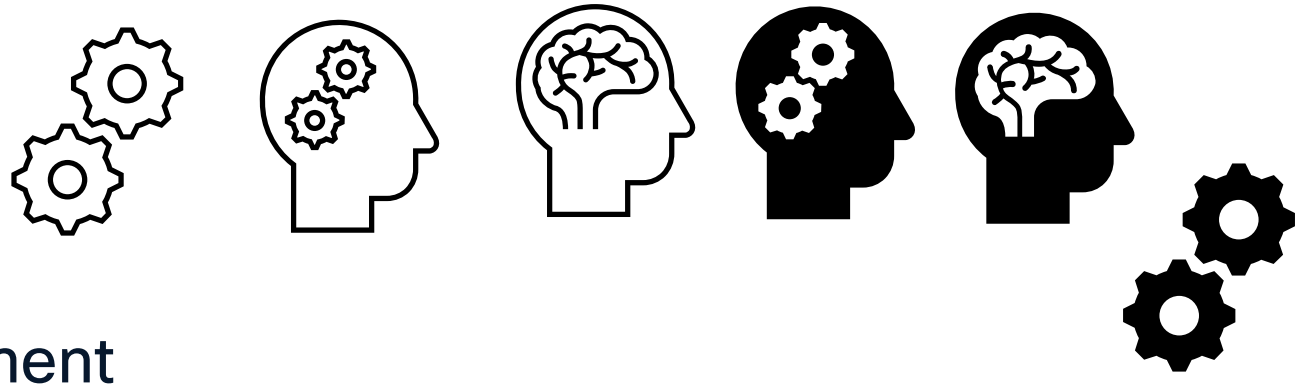
https://letsencrypt.org/2025/02/20/first-short-lived-cert-issued/

Demo – Let's Encrypt for Newbies

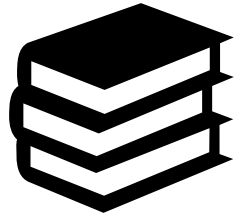
- Create a new Let's Encrypt Account
- Generate a new Let's Encrypt Certificate

Wrap this party up!!!

Recap



- Build your own Execution Environment
- Build Custom Credentials for your AuthC needs
- Certificates are great Automation Use Case



- [x] Use AWX for your basic and for your advanced ansible playbooks
- [x] Store your automation credentials securely in a External Vault
- [x] Create Custom Credential Types when more than a ‘password’ is required
- [x] Configure a credential (if req’d)
  Create a Job Tempate
- [x] Run the Job
  Monitor the output
  Schedule the job to run again!

Complete your session evaluations

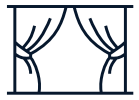
Complete a minimum of 4 session surveys and the Overall Event Survey to be entered in a drawing to win 1 of 5 full conference passes to Cisco Live 2026.

Earn 100 points per survey completed and compete on the Cisco Live Challenge leaderboard.

Level up and earn exclusive prizes!

Complete your surveys in the Cisco Live mobile app.

Continue your education



Visit the Cisco Showcase for related demos

Book your one-on-one Meet the Engineer meeting

Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs

Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

Contact me at: Webex

Thank you

cisco Live !


CISCO