Darden - Automation Consumer Lab

Summary of Exercises

Each student will have their own Ansible Automation Platform Controller environment that is able to manage 2 nodes. A Red Hat Enterprise Linux server (node1) and a Windows 2016 server (win1). For the purposes of this exercise, we will only be focusing on managing/automating the win1 node. The primary goal is to get students familiar with how to navigate within Automation Controller in order to execute jobs/playbooks and review output.

Note: the node1 and win1 instances do not have a lot of resources, so runtimes are a little longer than expected.

Exercise 1 - Accessing Your Environment

Go to: https://play.instruqt.com/embed/redhat/tracks/product-demos-dev
Short URL: https://red.ht/3FbQJ2I

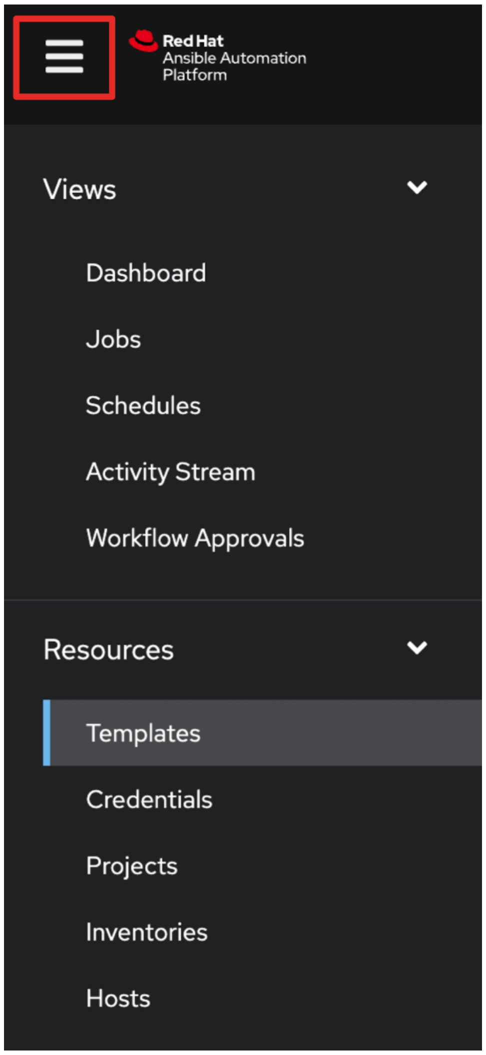
Launch Lab

- Click the green "Launch" button (wait approx 6 minutes)
- Click the green "Start" button

Log in to Automation Controller

USERNAME	**admin**
PASSWORD	**ansible123!**

Exercise 2 - Explore Automation Controller Resources



Automation Controller **resources** consist of Credentials, Projects, Inventories, and Templates. You can jump to available resources in the left hand navigation.

Tip: Left hand navigation may be hidden. Click on the 3 bars in the top menu to expand it (see Red Box in image to the left). Read descriptions below as you Click on each of the resources to explore and see what options they provide.

As an Automation Consumer, you will not need to modify any of these, it is just to provide you with the anatomy of automation.

- Projects → Review: Ansible Official Demo Project

  Defines a source code repository such as github, gitlab, or Azure Devops (to name a few). This is where the playbooks that have been written live and are version controlled.

- Credentials → Review: Controller Credential

  Represent all the secrets needed throughout the automation process. Examples include credentials to access:
  - your playbook in the git repository
  - the node/server you are automating
  - cloud providers' api's for provisioning
  - And much more…

- Inventories → Review: Workshop Inventory

  A manifest of nodes within your environment that you intend to automate. Inventories will contain hosts and/or groups of hosts, so you can target either one node, a group of similar/related nodes, or the entire inventory.

- Templates → Review: SETUP

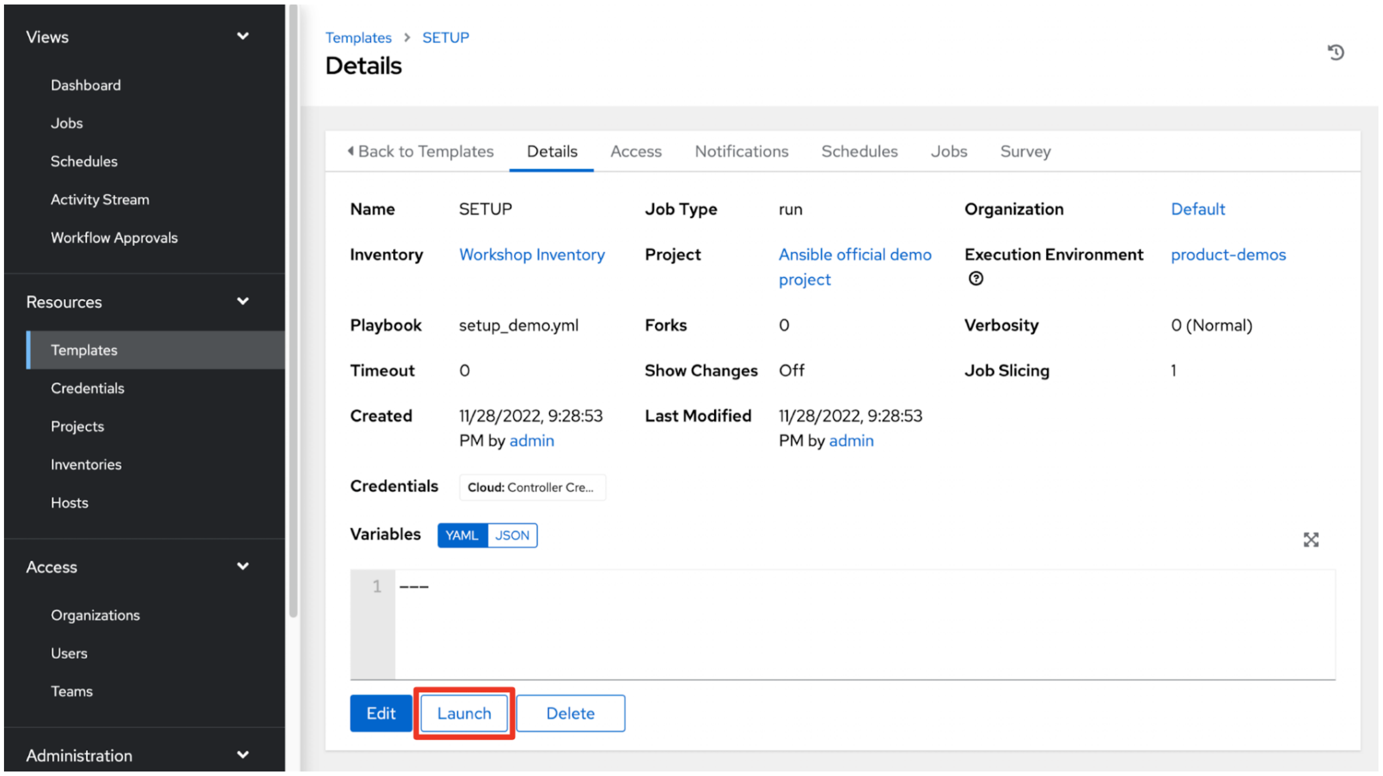
  Templates are used to run Ansible Playbooks that have been written by content creators and domain experts. However, these templates need a lot of information in order to launch the playbooks. That is where the other resources we just reviewed come in. In order to launch a Template, you will need to add the other resources to the job template definition. The fields collected in the job template should look very familiar (projects, credentials, inventories, and more).

  Note: Don't exit out of the SETUP Job Template, as we will be using it for exercise 3 below.

Exercise 3 - Launch a Job Template (2.5 min runtime)

As an End User, or Consumer of Ansible, this is where you will be spending all of your time. You will not need to view or edit any of the other resources other than Templates. All Job templates (single playbook) and Workflow Job Templates (series of playbooks) are launched with the simple click of a button.

Click the Launch button in the SETUP job template:



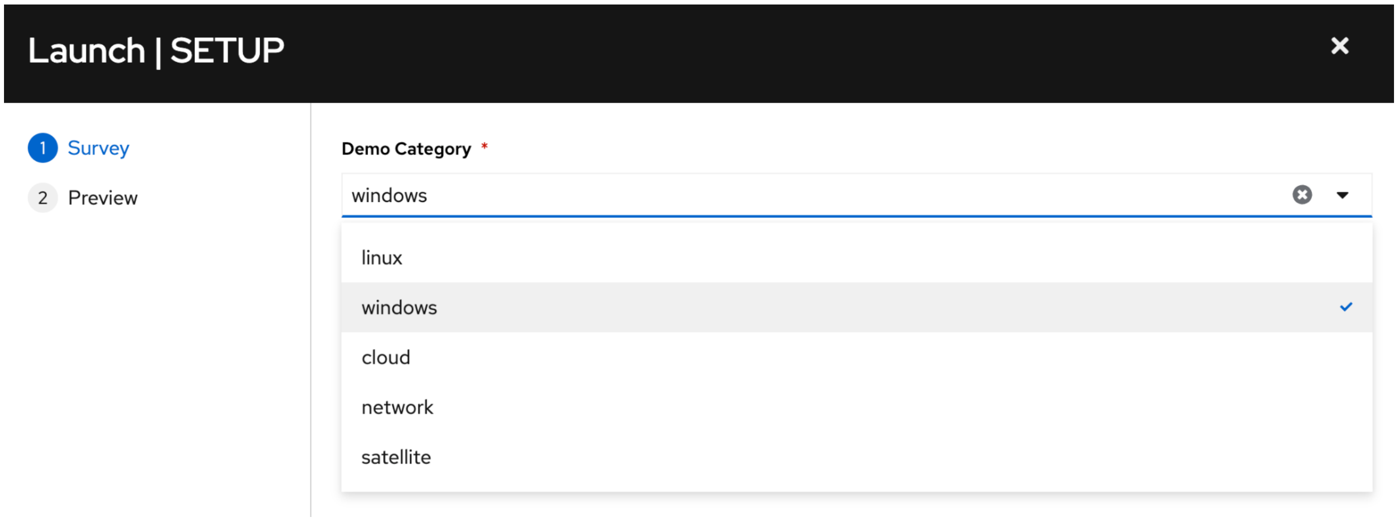
Name	SETUP	Job Type	run	Organization	Default
Inventory	Workshop Inventory	Project	Ansible official demo project	Execution Environment	product-demos
Playbook	setup_demo.yml	Forks	0	Verbosity	0 (Normal)
Timeout	0	Show Changes	Off	Job Slicing	1
Created	11/28/2022, 9:28:53 PM by admin	Last Modified	11/28/2022, 9:28:53 PM by admin		
Credentials	Cloud: Controller Cre...				

```
---
```

Select **Windows** for the Demo Category

Playbooks can use variables supplied via a form (called a Survey) that can dynamically define how the playbook is run. In this particular case, we are executing a Job Template that automates the build of many more job templates for you to play around with. Depending on which Category you select, will dictate which job templates are added to the Automation Controller.

Go ahead and select **windows** in the drop down listbox, then click the **Next** button

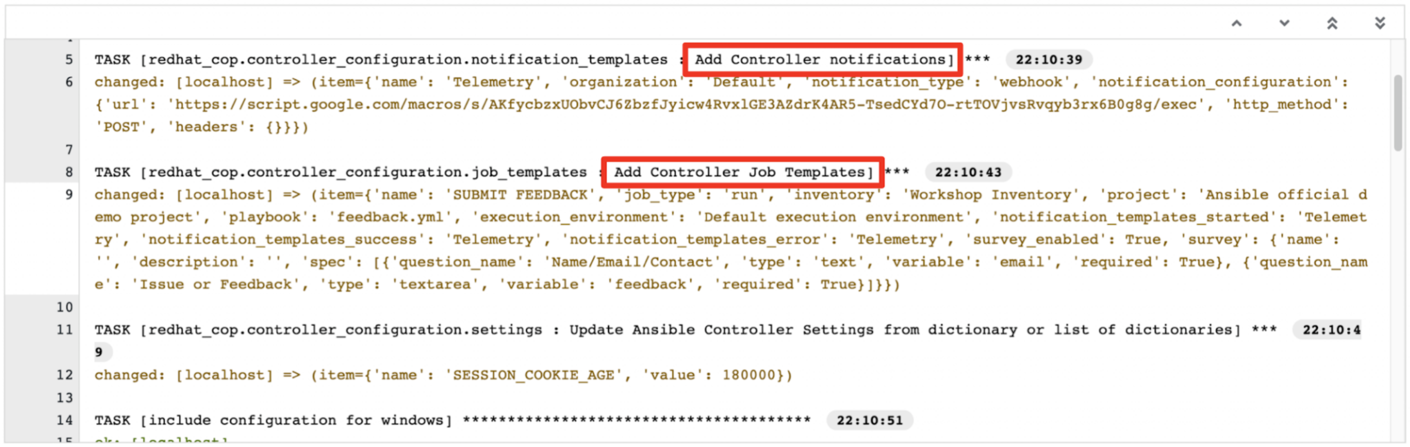


On the subsequent preview screen you are given the opportunity to review the Job Template configuration. Once settings are confirmed, click the **Launch** button

Review the Job Template output

On the subsequent screen you will see the output of the playbook. This is critical from an auditing perspective. Automaton Controller centrally logs all the jobs that have been executed with key information such as who launched it, when it was launched, the duration, and the exact changes that were made on each of the systems that were targeted.

Scroll through the output to see the user-friendly task definitions that explain what the task is doing (highlighted by the red boxes).



```
TASK [redhat_cop.controller_configuration.notification_templates : Add Controller notifications] ***  22:10:39
changed: [localhost] => (item={'name': 'Telemetry', 'organization': 'Default', 'notification_type': 'webhook', 'notification_configuration':
{'url': 'https://script.google.com/macros/s/AKfycbzxUObvCJ6ZbzfJyicw4RvxlGE3AZdrK4AR5-TsedCYd7O-rtTOVjvsRvqyb3rx6B0g8g/exec', 'http_method':
'POST', 'headers': {}}})

TASK [redhat_cop.controller_configuration.job_templates : Add Controller Job Templates] ***  22:10:43
changed: [localhost] => (item={'name': 'SUBMIT FEEDBACK', 'job_type': 'run', 'inventory': 'Workshop Inventory', 'project': 'Ansible official d
emo project', 'playbook': 'feedback.yml', 'execution_environment': 'Default execution environment', 'notification_templates_started': 'Telemet
ry', 'notification_templates_success': 'Telemetry', 'notification_templates_error': 'Telemetry', 'survey_enabled': True, 'survey': {'name':
'', 'description': '', 'spec': [{'question_name': 'Name/Email/Contact', 'type': 'text', 'variable': 'email', 'required': True}, {'question_nam
e': 'Issue or Feedback', 'type': 'textarea', 'variable': 'feedback', 'required': True}]}})

TASK [redhat_cop.controller_configuration.settings : Update Ansible Controller Settings from dictionary or list of dictionaries] ***  22:10:4
9
changed: [localhost] => (item={'name': 'SESSION_COOKIE_AGE', 'value': 180000})

TASK [include configuration for windows] ****************************************  22:10:51
ok: [localhost]
```

See the Fruits of our button clicks:

- Go back to Templates (under **Resources** in the left hand navigation)
- See all the recently added Job Templates with the **WINDOWS /…** prefix that were built to help you automate your windows environment.

Exercise 4 - Automate Windows with Job Templates

Now that we can play around with these new job templates and see how easy it is to perform some otherwise complex and time consuming tasks. To top it all off, because you are only having to click buttons and answer simple questions there is very little margin for human error..

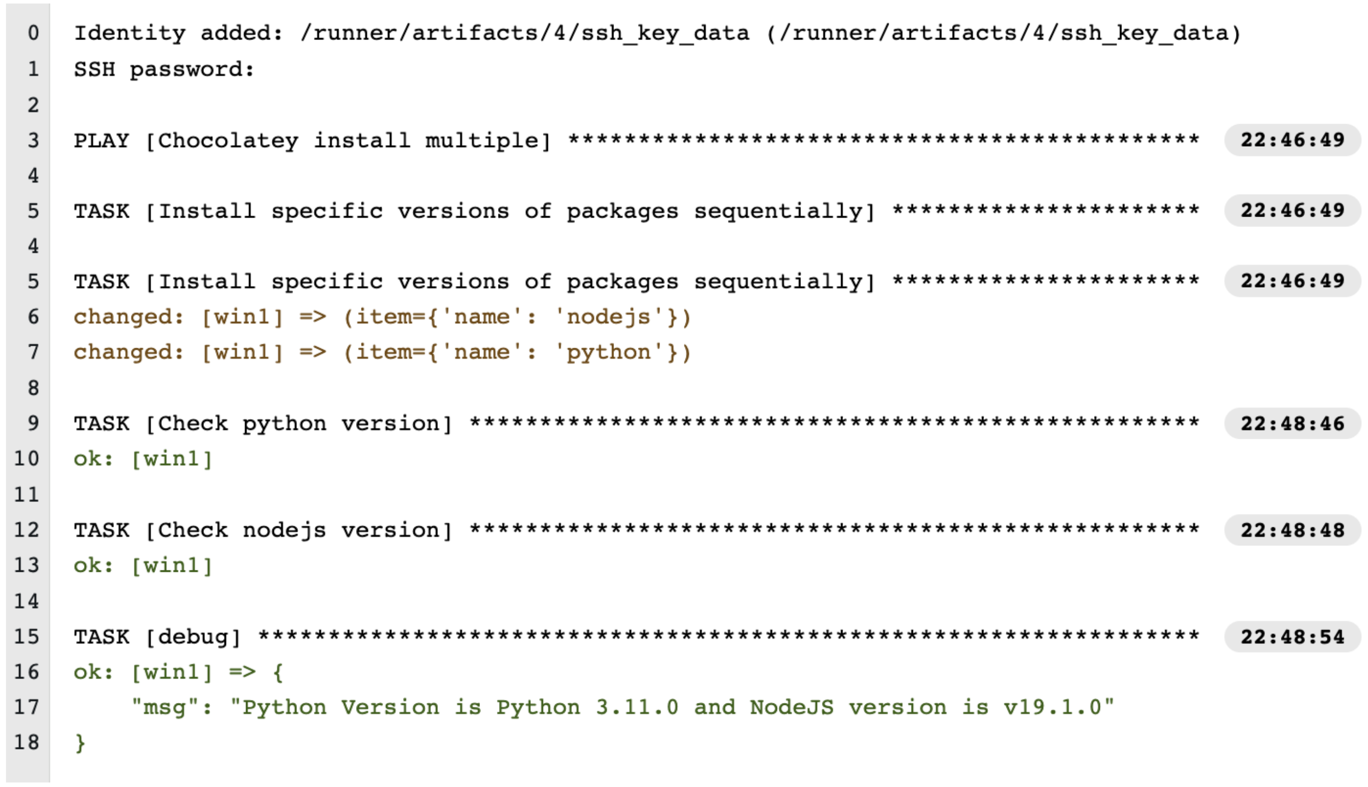
- **Install Software (2.5 min runtime)**

  Got to **Templates**

  Click the icon next to **WINDOWS / Chocolatey Install Multiple**

  Type **win1** in the survey → Click **Next** → Click **Launch**

  Review the output (you just installed nodejs and python on Win1):



```
Identity added: /runner/artifacts/4/ssh_key_data (/runner/artifacts/4/ssh_key_data)
SSH password:

PLAY [Chocolatey install multiple] *********************************************  22:46:49

TASK [Install specific versions of packages sequentially] **********************  22:46:49

TASK [Install specific versions of packages sequentially] **********************  22:46:49
changed: [win1] => (item={'name': 'nodejs'})
changed: [win1] => (item={'name': 'python'})

TASK [Check python version] ****************************************************  22:48:46
ok: [win1]

TASK [Check nodejs version] ****************************************************  22:48:48
ok: [win1]

TASK [debug] *******************************************************************  22:48:54
ok: [win1] => {
    "msg": "Python Version is Python 3.11.0 and NodeJS version is v19.1.0"
}
```

  Note: You also gathered information about the versions that were installed.

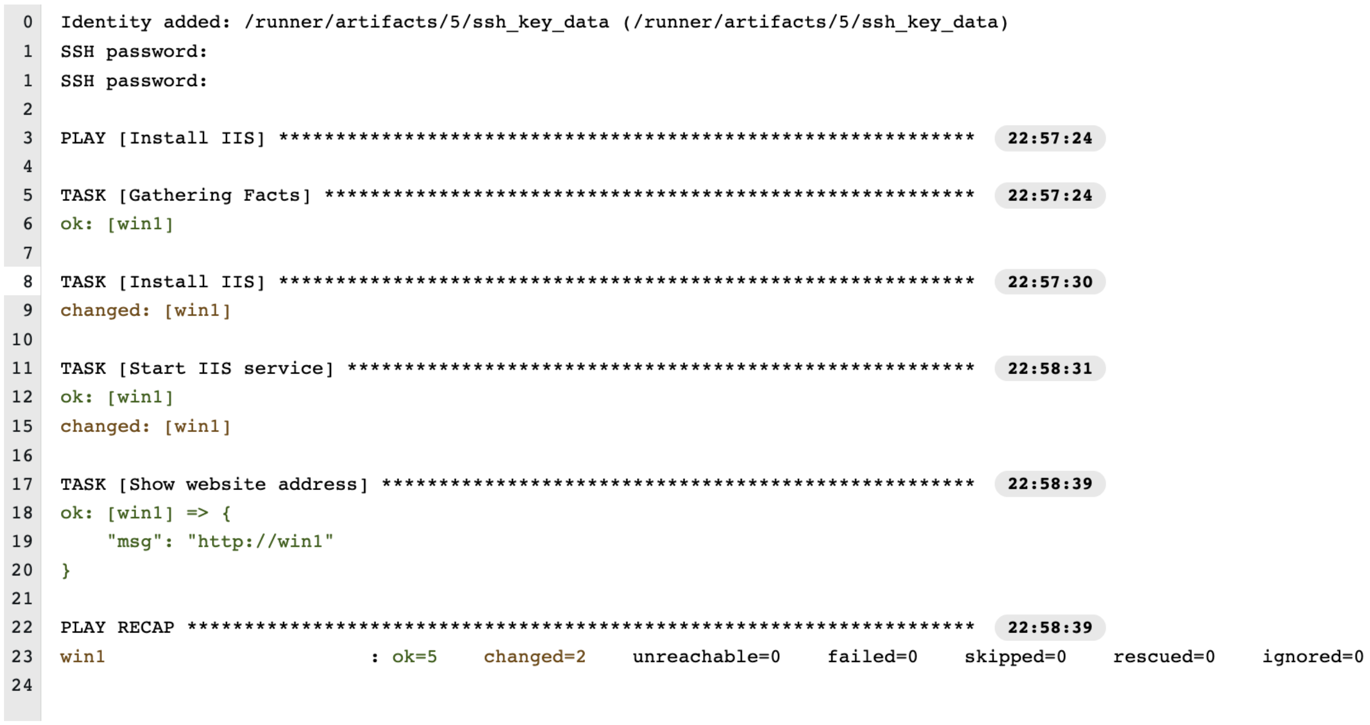
- **Deploy a Website (7 min runtime)**

  Got to **Templates**

Click the icon next to **WINDOWS / Install IIS**

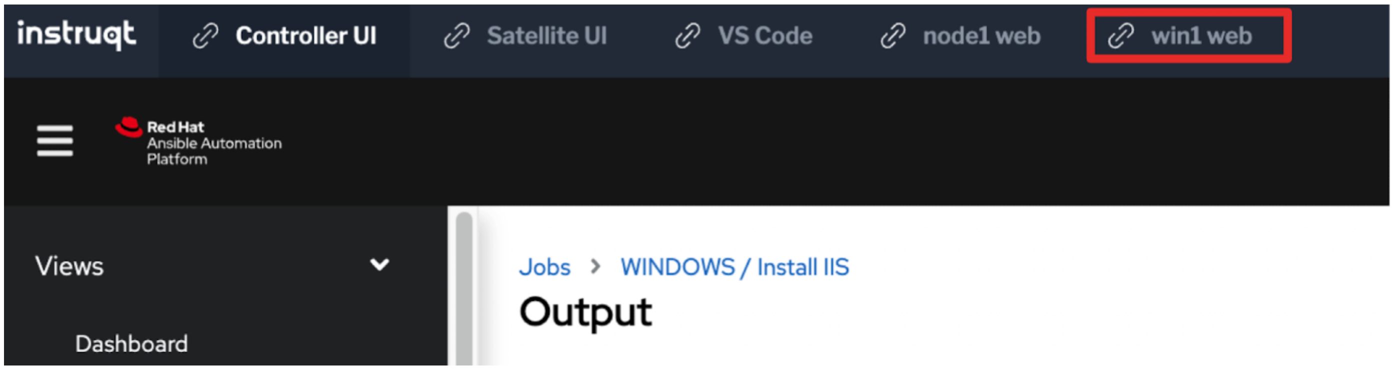
Type **win1** for Server Name **&** <whatever you like> for web content → Click **Next** → Click **Launch**

Review the output (you just installed IIS, started the service and deployed a website on wiin1):

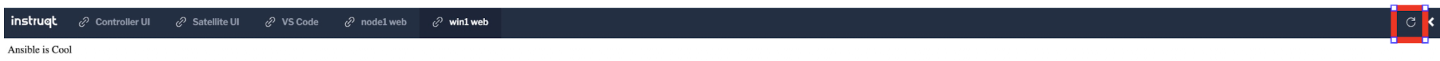


```
Identity added: /runner/artifacts/5/ssh_key_data (/runner/artifacts/5/ssh_key_data)
SSH password:
SSH password:

PLAY [Install IIS] *************************************************************  22:57:24

TASK [Gathering Facts] *********************************************************  22:57:24
ok: [win1]

TASK [Install IIS] *************************************************************  22:57:30
changed: [win1]

TASK [Start IIS service] *******************************************************  22:58:31
ok: [win1]
changed: [win1]

TASK [Show website address] ****************************************************  22:58:39
ok: [win1] => {
    "msg": "http://win1"
}

PLAY RECAP *********************************************************************  22:58:39
win1                       : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

Check out your new custom website that you built (click on the **win1_web** tab at the top of the screen):



You may need to click the **refresh icon** on the top right of the screen:



Ansible is Cool

- **Update password settings (1 min runtime)**

Click the icon next to **WINDOWS / Configuring Password Requirements**

Type **win1** for Server Name → Click **Next** → Click **Launch**

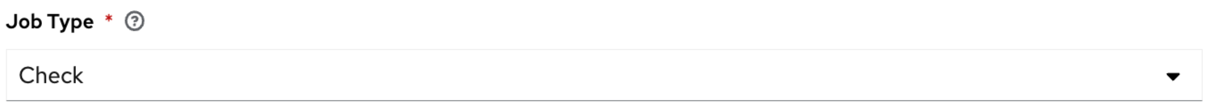
Review the output (you just installed IIS, started the service and deployed a website on win1):

- **Check for updates on a windows server (7 min runtime)**

  Go to **Templates**

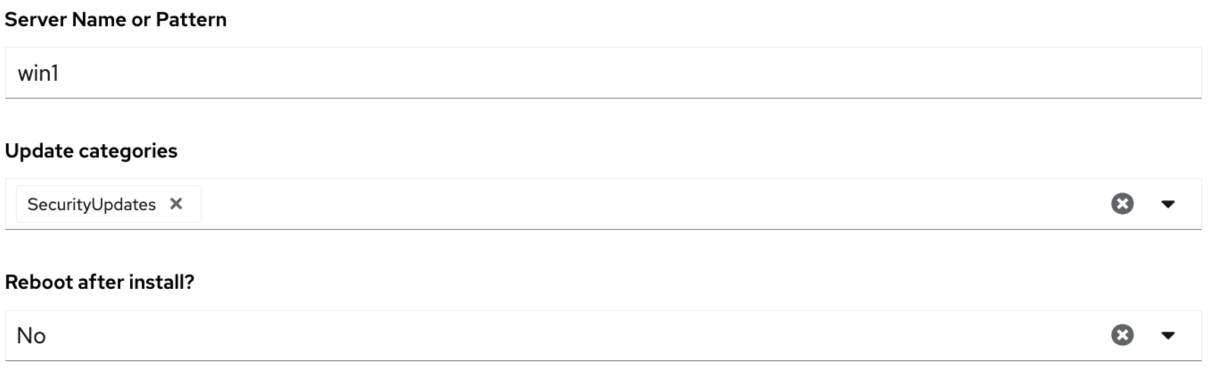
  Click the icon next to **WINDOWS / Patching**

  Select "Check" for Job Type → Next



Job Type *
Check

  Complete Survey as follows → Next → Launch

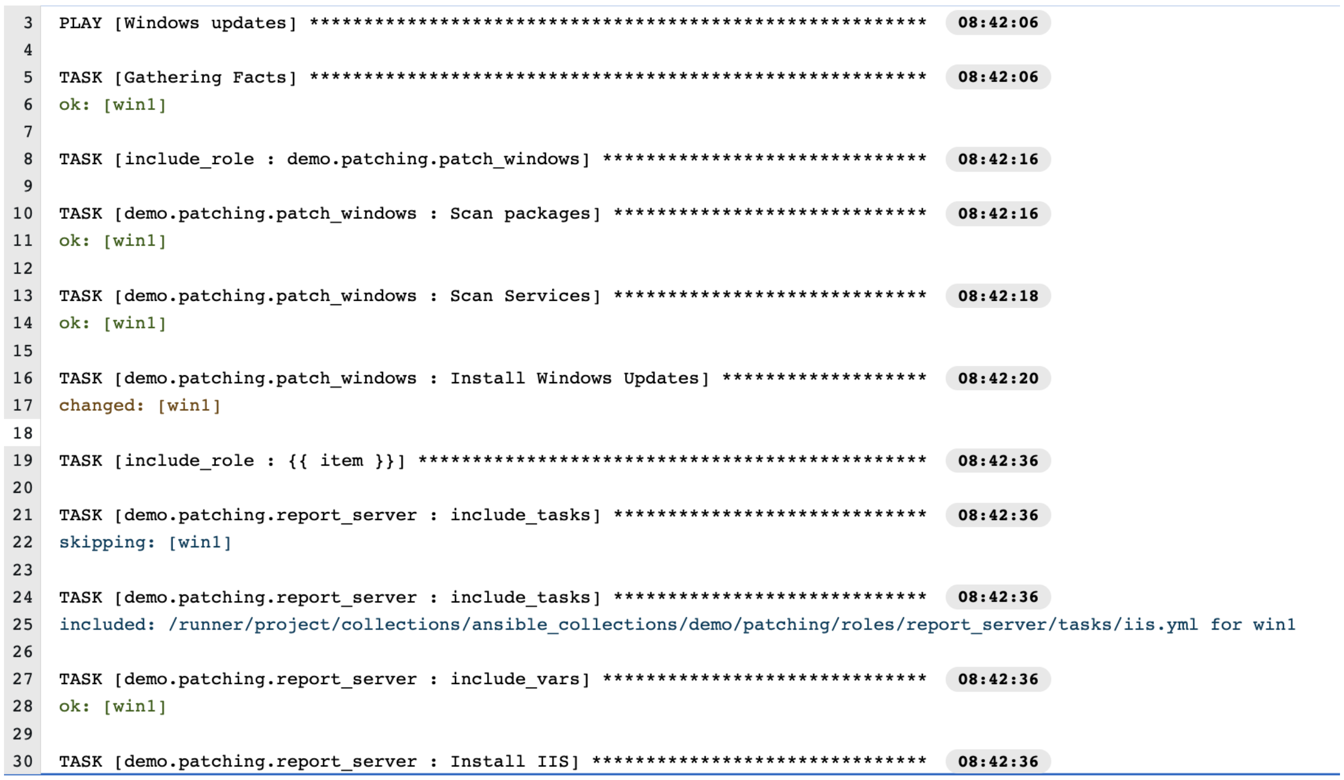


Server Name or Pattern
win1

Update categories
SecurityUpdates

Reboot after install?
No

  Review Output



```
PLAY [Windows updates] **********************************************************  08:42:06

TASK [Gathering Facts] **********************************************************  08:42:06
ok: [win1]

TASK [include_role : demo.patching.patch_windows] ******************************  08:42:16

TASK [demo.patching.patch_windows : Scan packages] *****************************  08:42:16
ok: [win1]

TASK [demo.patching.patch_windows : Scan Services] *****************************  08:42:18
ok: [win1]

TASK [demo.patching.patch_windows : Install Windows Updates] *******************  08:42:20
changed: [win1]

TASK [include_role : {{ item }}] ************************************************  08:42:36

TASK [demo.patching.report_server : include_tasks] *****************************  08:42:36
skipping: [win1]

TASK [demo.patching.report_server : include_tasks] *****************************  08:42:36
included: /runner/project/collections/ansible_collections/demo/patching/roles/report_server/tasks/iis.yml for win1

TASK [demo.patching.report_server : include_vars] ******************************  08:42:36
ok: [win1]

TASK [demo.patching.report_server : Install IIS] *******************************  08:42:36
```

  Note: as of November 29, 2022, The WINDOWS / Patching job template searches for updates that meet your criteria, but there is an error when creating the report that shows the patches. You will see an error and you will see it fail, but if you review the output you will see that the updates did actually occur. As an automation consumer, this is when you would reach out to the AAP Operator in your organization. They will love the fact that this is all centrally logged in great detail, so they will be able to go in and troubleshoot the error and get it fixed.