

Ansible Tower on the AWS Cloud

Quick Start Reference Deployment

Tony Vattathil

Solutions Architect, AWS Quick Start Reference Team

April 2016

This guide is also available in HTML format at
<http://aws.amazon.com/quickstart/ansible-tower/>.



Contents

Overview	3
Ansible Tower on AWS	3
Quick Links	4
Cost and Licenses	4
Architecture	5
AWS Services	6
Ansible Tower Installation	6
Deployment Steps	7
What We'll Cover	7
Step 1. Prepare an AWS Account	8
Step 2. Launch the Ansible Tower Stack	11
Step 3. Create a User Account for Ansible Tower	12
Step 4. Get a Trial License for Ansible Tower	14
Step 5. Configure and Manage EC2 Instances in Tower	18
Configuring Ansible Tower with EC2 Integration	18
Discovering and Managing EC2 Instances in Ansible Tower	21
Adding Other Managed Instances	27
Troubleshooting	28
Security	29
Additional Resources	29
Send Us Feedback	30
Document Revisions	30

About This Guide

This Quick Start reference deployment guide discusses the steps for deploying and testing Ansible Tower on the Amazon Web Services (AWS) cloud. It provides links for viewing and launching the [AWS CloudFormation](#) template that automates the Ansible Tower deployment, and creates Ansible-aware Amazon Elastic Compute Cloud (Amazon EC2) instances. The guide also explains how you can configure Ansible Tower to pull EC2 instances and their metadata into the Ansible Tower dashboard.

The guide is for IT infrastructure architects, administrators, and DevOps professionals who are planning to implement Ansible Tower to manage their AWS compute resources.

AWS OpsWorks option This Quick Start is for customers who want to run and manage an Ansible Tower infrastructure on AWS. However, we recommend that you also take a look at AWS OpsWorks, which is a configuration management service provided by AWS, to determine if it's more suitable for your needs. AWS OpsWorks helps you configure and operate applications of all types and sizes. You can define the application's architecture and the specification of each component, including package installation, software configuration, and resources such as storage. For more information, see the [AWS OpsWorks User Guide](#).

[Quick Starts](#) are automated reference deployments for key workloads on the AWS cloud. Each Quick Start launches, configures, and runs the AWS compute, network, storage, and other services required to deploy a specific workload on AWS, using AWS best practices for security and availability.

Overview

Ansible Tower on AWS

Ansible is an IT DevOps tool that automates provisioning, configuration management, application deployment, intra-service orchestration, continuous delivery, and many other IT processes.

Ansible is designed for multi-tier deployments. Instead of managing systems individually, it models your IT infrastructure by describing the inter-relationships among all your systems.

Ansible Tower is a web-based user interface for Ansible. Its visual dashboard lets you schedule and deploy Ansible playbooks, and provides centralized logging, auditing, and system tracking.

A key advantage to Ansible over other automation engines is that it uses no agents and no additional custom security infrastructure, which simplifies deployment. Ansible uses a very simple, human-readable language called *YAML* for Ansible playbooks, to manage configuration, deployment, and orchestration tasks. Ansible works by connecting to your nodes and running small programs, called *Ansible modules*, to configure the resource for your system. Ansible executes these modules over Secure Shell (SSH) by default, and removes them when finished.

Quick Links

If you have an AWS account and you're already familiar with AWS and Ansible Tower, you can launch the Quick Start to build the architecture shown in [Figure 1](#). You can choose to deploy Ansible Tower on an Amazon Elastic Compute Cloud (Amazon EC2) instance that is running CentOS 7 or Red Hat Enterprise Linux (RHEL) 7. The deployment takes approximately 20 minutes. If you're new to AWS or Ansible Tower, please review the implementation details and follow the [step-by-step instructions](#) provided later in this guide to launch the Quick Start.

**Launch
Quick Start (CentOS)**

**Launch
Quick Start (RHEL)**

If you want to take a look under the covers, you can view the AWS CloudFormation template that automates the deployment on CentOS 7 or RHEL 7. The default configuration deploys two servers that use the t2.medium instance type by default. You can customize the template during launch, or download and extend it for other projects.

View template (CentOS)

View template (RHEL)

Cost and Licenses

You are responsible for the cost of the AWS services used while running this Quick Start reference deployment. There is no additional cost for using the Quick Start. See the pricing pages for each AWS service you will be using or the [AWS Simple Monthly Calculator](#) for full details. A brief cost estimate is also displayed in the final AWS CloudFormation screen before you create the stack.

This Quick Start deploys Ansible Tower on the AWS cloud along with Linux Ansible client. Ansible Tower is subject to the [Ansible Software Subscription and Services Agreement](#). A free trial version of Ansible Tower is available for managing up to 10 hosts. After you deploy the Quick Start, you can follow the step-by-step instructions in this guide to acquire the 10-host trial license.

Architecture

Deploying this Quick Start with the **default parameters** builds the following environment in the AWS cloud.

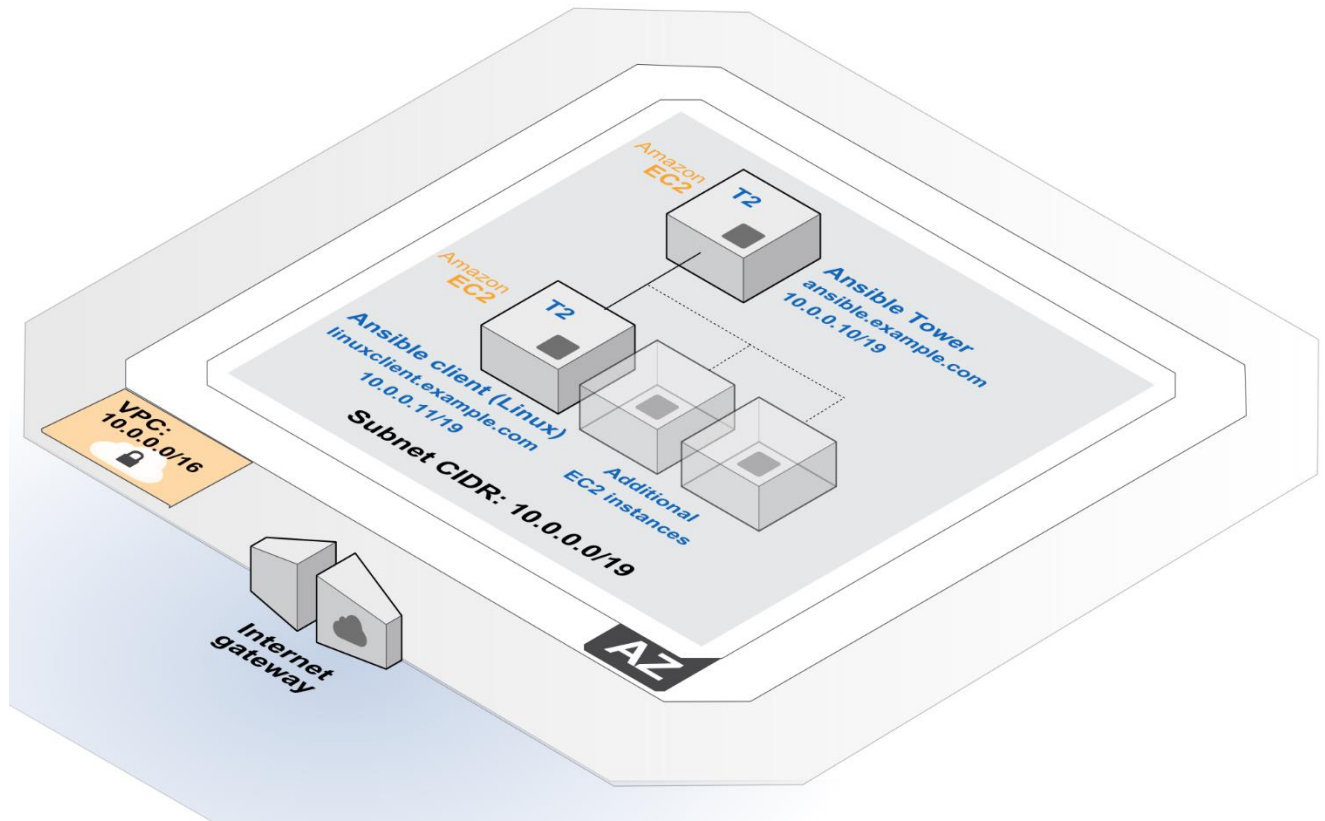


Figure 1: Quick Start Architecture for Ansible Tower on AWS

This Quick Start deploys the resources shown in Figure 1 and uses them as follows:

- An Amazon VPC is created in the region you choose when you launch the stack. A single, public VPC subnet is created in an Availability Zone.
- One Linux client instance is deployed into the VPC subnet.
- One Ansible Tower instance is deployed into the VPC subnet.

AWS Services

The core AWS components used by this Quick Start include the following AWS services. (If you are new to AWS, see the [Getting Started section](#) of the AWS documentation.)

- [Amazon EC2](#) – The Amazon Elastic Compute Cloud (Amazon EC2) service enables you to launch virtual machine instances with a variety of operating systems. You can choose from existing Amazon Machine Images (AMIs) or import your own virtual machine images.
- [Amazon VPC](#) – The Amazon Virtual Private Cloud (Amazon VPC) service lets you provision a private, isolated section of the AWS cloud where you can launch AWS services and other resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.
- [AWS CloudFormation](#) – AWS CloudFormation gives you an easy way to create and manage a collection of related AWS resources, and provision and update them in an orderly and predictable way. You use a template to describe all the AWS resources (for example, Amazon EC2 instances) that you want. You don't have to individually create and configure the resources or figure out dependencies—AWS CloudFormation handles all of that.
- [IAM](#) – AWS Identity and Access Management (IAM) enables you to securely control access to AWS services and resources for your users. With IAM, you can centrally manage users, security credentials such as access keys, and permissions that control which AWS resources users can access.

Ansible Tower Installation

This Quick Start deploys the latest version of Ansible Tower on an EC2 instance that is running CentOS 7 or Red Hat Enterprise Linux (RHEL) 7. The installation is automated with a user data script that executes when the instance is launched via AWS CloudFormation. Ansible Tower installation files are installed directly from Ansible's release server.

In addition to installing Ansible Tower, the Quick Start also deploys a Linux client into the Amazon VPC. The client is tagged with the key **Tower**. After you deploy the Quick Start, you'll use this key to identify and manage the client in Ansible Tower. We'll provide step-by-step instructions for doing that in [step 5](#) of the deployment section.

Deployment Steps

The AWS CloudFormation template provided with this Quick Start bootstraps the AWS infrastructure and automates the deployment of Ansible Tower on the AWS cloud from scratch. Follow the step-by-step instructions in this section to set up your AWS account, customize the template, and deploy the software into your account.

What We'll Cover

The procedure for deploying and testing Ansible Tower on AWS consists of the following steps. For detailed instructions, follow the links for each step.

[Step 1. Prepare an AWS account](#)

- Sign up for an AWS account, if you don't already have one.
- Choose the region where you want to deploy the stack on AWS.
- Create a key pair in the region.
- Review account limits for Amazon EC2 instances, and request a limit increase, if needed.

[Step 2. Launch the stack](#)

- Launch the AWS CloudFormation template into your AWS account.
- Enter values for all parameters that require input.
- Review other template parameters, and customize their values if necessary.

[Step 3. Create a user account](#)

- Create a user account for Ansible Tower, and assign it the IAM PowerUserAccess policy. Alternatively, you can use an existing administrator account.

[Step 4. Get an Ansible Tower trial license](#)

- Connect to your Ansible Tower via a web browser and follow the steps to license Ansible Tower.

[Step 5. Configure and manage EC2 instances in Ansible Tower](#)

- Configure Ansible Tower with EC2 integration.
- Integrate AWS compute resources into the Ansible Tower inventory.
- Add other managed instances that you can view and configure in Ansible Tower.

Step 1. Prepare an AWS Account

1. If you don't already have an AWS account, create one at <http://aws.amazon.com> by following the on-screen instructions. Part of the sign-up process involves receiving a phone call and entering a PIN using the phone keypad.
2. Use the region selector in the navigation bar to choose the Amazon EC2 region where you want to deploy Ansible Tower on AWS.

Amazon EC2 locations are composed of *Regions* and *Availability Zones*. Regions are dispersed and located in separate geographic areas.

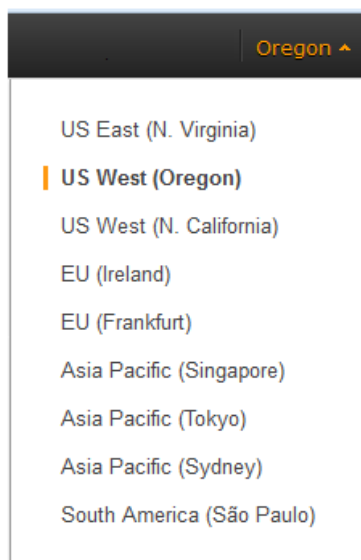


Figure 2: Choosing an Amazon EC2 Region

Tip Consider choosing a region closest to your data center or corporate network to reduce network latency between systems running on AWS and the systems and users on your corporate network.

3. Create a [key pair](#) in your preferred region. To do this, in the navigation pane of the Amazon EC2 console, choose **Key Pairs**, **Create Key Pair**, type a name, and then choose **Create**.

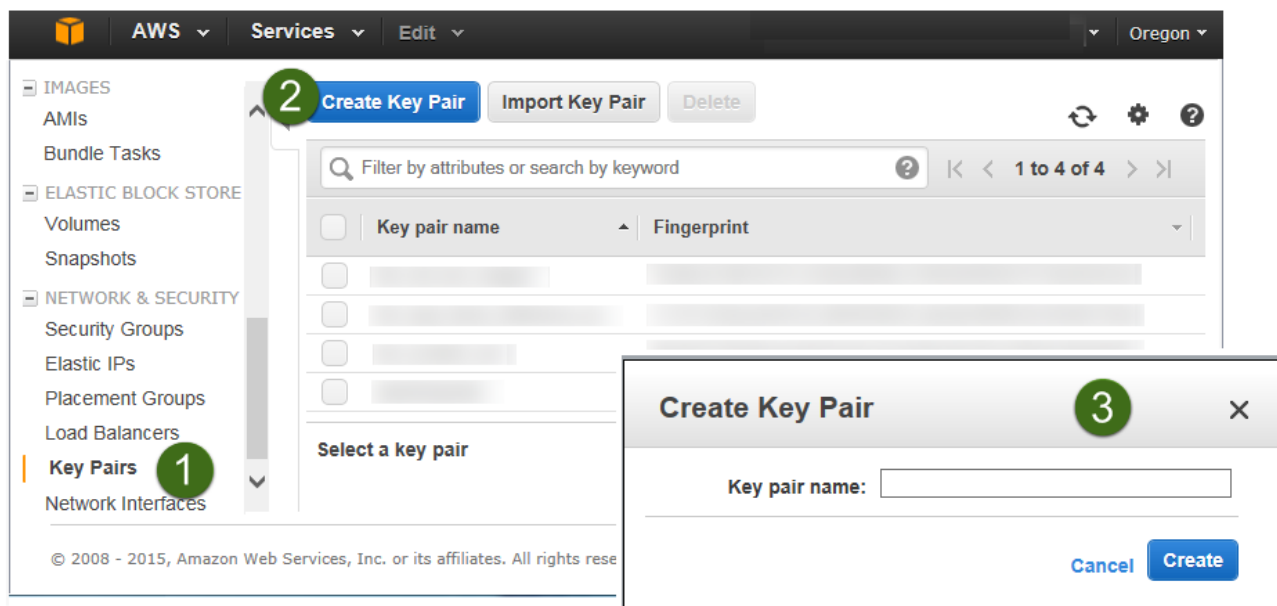


Figure 3: Creating a Key Pair

Amazon EC2 uses public-key cryptography to encrypt and decrypt login information. To be able to log in to your instances, you must create a key pair. On Linux, we use the key pair to authenticate SSH login.

4. If necessary, [request a service limit increase](#) for the Amazon EC2 **t2.medium** instance type. To do this, in the AWS Support Center, choose **Create Case, Service Limit Increase, EC2 instances**, and then complete the fields in the limit increase form. The current default limit for this instance type is 20 instances. (You can also choose a different instance type by customizing AWS CloudFormation parameters, as explained in [step 2](#).)

You might need to request an increase if you already have an existing deployment that uses this instance type, and you think you might exceed the default limit with this reference deployment. It might take a few days for the new service limit to become effective. To learn more, see [Amazon EC2 Service Limits](#) in the AWS documentation.

The screenshot displays the AWS Support Center interface for creating a new case. The left sidebar contains a navigation menu with 'Dashboard', 'Create Case' (highlighted with a green circle 1), and 'Case History'. The main content area is titled 'Create Case' and shows the 'Basic Support Plan' selected. The form includes the following elements:

- Name:** A text input field.
- Account:** A text input field.
- CC:** A text input field for contact information, with a note: 'Required for IAM users; use commas or semicolons to separate email addresses'.
- Regarding*:** Radio buttons for 'Account and Billing Support', 'Service Limit Increase' (selected, marked with a green circle 2), and 'Technical Support' (with a note: 'Unavailable under the Basic Support Plan').
- Limit Type*:** A dropdown menu set to 'EC2 Instances' (marked with a green circle 3).
- Request 1:** A section containing:
 - Region*:** A dropdown menu set to 'US West (Oregon)'.
 - Primary Instance:** A dropdown menu set to 'c3.8xlarge'.
 - Type*:** A dropdown menu set to 'Instance Limit'.
 - Limit*:** A dropdown menu set to 'Instance Limit'.
 - New limit value*:** A text input field set to '25' (marked with a green circle 4).
- Add another request:** A button at the bottom of the form.

Figure 4: Requesting a Service Limit Increase

Step 2. Launch the Ansible Tower Stack

The automated AWS CloudFormation template provided with this Quick Start deploys Ansible Tower into an Amazon VPC. Please make sure that you've completed the previous steps before launching the stack.

1. Launch the AWS CloudFormation template into your AWS account for deployment on CentOS 7 or RHEL 7.

The template is launched in the US West (Oregon) Region by default. You can change the region by using the region selector in the navigation bar.

This stack takes approximately 20 minutes to create.

**Launch
Quick Start (CentOS)**

**Launch
Quick Start (RHEL)**

Note You are responsible for the cost of the AWS services used while running this Quick Start reference deployment. There is no additional cost for using this Quick Start.

You can also download the [CentOS template](#) or [RHEL template](#) to use it as a starting point for your own implementation.

2. On the **Select Template** page, keep the default URL for the AWS CloudFormation template, and then choose **Next**.
3. On the **Specify Details** page, review the parameters for the template. These are described in the following table. Provide values for the **AnsibleAdminPassword**, **DatabaseAdminPassword**, and **KeyPairName** parameters. These parameters require your input. For all other parameters, the template provides default settings that you can customize.

Important Make a note of the password you provide for the **AnsibleAdminPassword** parameter. You will need this password in the steps to follow.

Parameter name	Default	Description
AnsibleAdminPassword	<i>Requires input</i>	Password for the Ansible Tower administrator account. This must be at least 8 characters, including letters, numbers, and symbols, and must contain at least one uppercase letter (e.g., An\$ib13ChgMe).
AnsibleClientLinuxIP	10.0.0.11	IP address of a Linux instance that can be managed by Ansible.
AnsibleTowerInstanceType	t2.medium	Amazon EC2 instance type for Ansible Tower.

Parameter name	Default	Description
AnsibleTowerIP	10.0.0.10	Private IP of your Ansible Tower.
DatabaseAdminPassword	<i>Requires input</i>	Password for the Ansible database administrator account. This must be at least 8 characters, including letters, numbers, and symbols, and must contain at least one uppercase letter. All database resources (e.g., Redis, Postgres) will use this password.
KeyPairName	<i>Requires input</i>	Public/private key pair, which enables you to connect securely to your instance after it launches. When you created an AWS account, this is the key pair you created in your preferred region.
LinuxClientInstanceType	t2.medium	Amazon EC2 instance type for the Linux client.
RemoteAdminCIDR	0.0.0.0/0	CIDR block or IP address for SSH (e.g., 1.1.1.1/32).
SubnetCIDR	10.0.0.0/19	CIDR block for the public subnet in the Amazon VPC where Ansible Tower will be deployed.
VPCCIDR	10.0.0.0/16	CIDR block for the Amazon VPC.

Note You can also download the template (for [CentOS](#) or [RHEL](#)) and edit it to create your own parameters based on your specific deployment scenario.

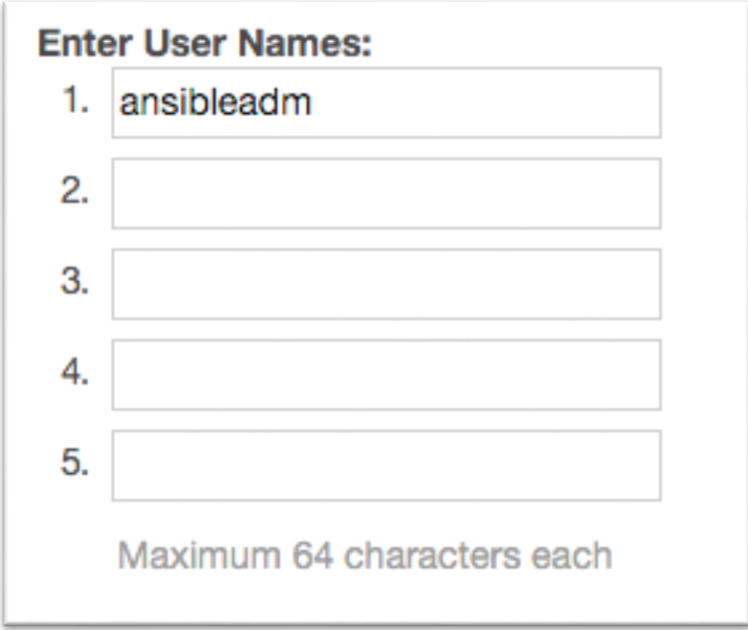
- On the **Options** page, you can [specify tags](#) (key-value pairs) for resources in your stack and [set additional options](#). When you're done, choose **Next**.
- On the **Review** page, review and confirm the settings. Under **Capabilities**, select the check box to acknowledge that the template will create IAM resources.
- Choose **Create** to deploy the stack.
- Monitor the status of the stack. When the status is **CREATE_COMPLETE**, the Ansible Tower stack is ready. The IP address is displayed in the **Outputs** tab for the stack, as shown later in Figure 8.

Important We recommend that you update the passwords for the administrator and database administrator accounts in accordance with your IT standards after stack creation is complete.

Step 3. Create a User Account for Ansible Tower

Let's create a new user called *ansibleadm* for Ansible Tower. This is an optional step—you can use an existing user account instead, if that account has the proper privileges. (At a minimum, you'll need read-only access to EC2 instances to pull in metadata.)

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Users**, and then choose **Create New Users**.
3. Type **ansibleadm**, and then choose **Create**.



Enter User Names:

1.
2.
3.
4.
5.

Maximum 64 characters each

Figure 5: Creating a User Account for Ansible Tower

4. Choose **Show User Security Credentials**. Copy or download the credentials to your workstation, and then choose **Close**.



ansibleadm

Access Key ID:

Secret Access Key:

Figure 6: Security Credentials for User Account

5. Choose **ansibleadm**, and then choose the **Permissions** tab so you can assign IAM roles to this user.

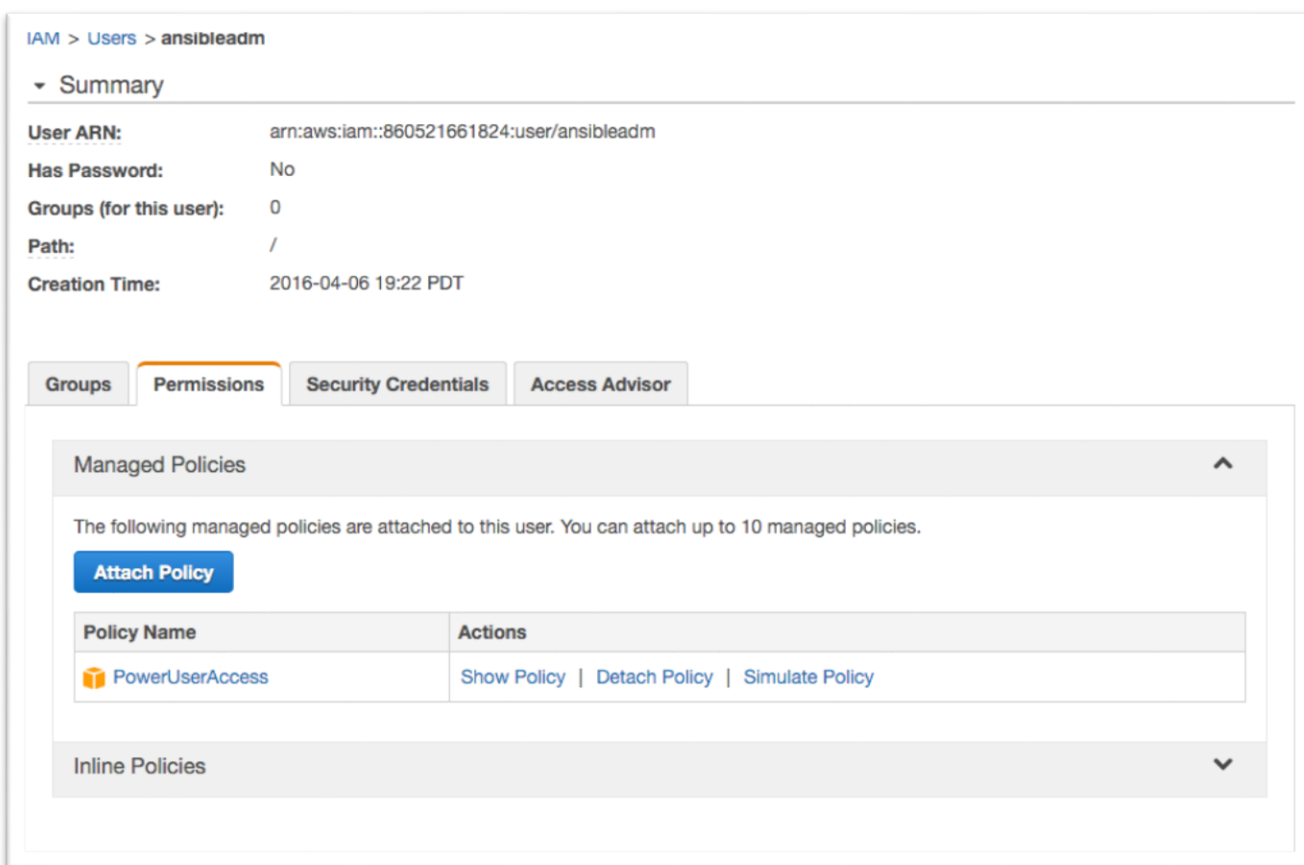


Figure 7: Assigning IAM Permissions to the User Account

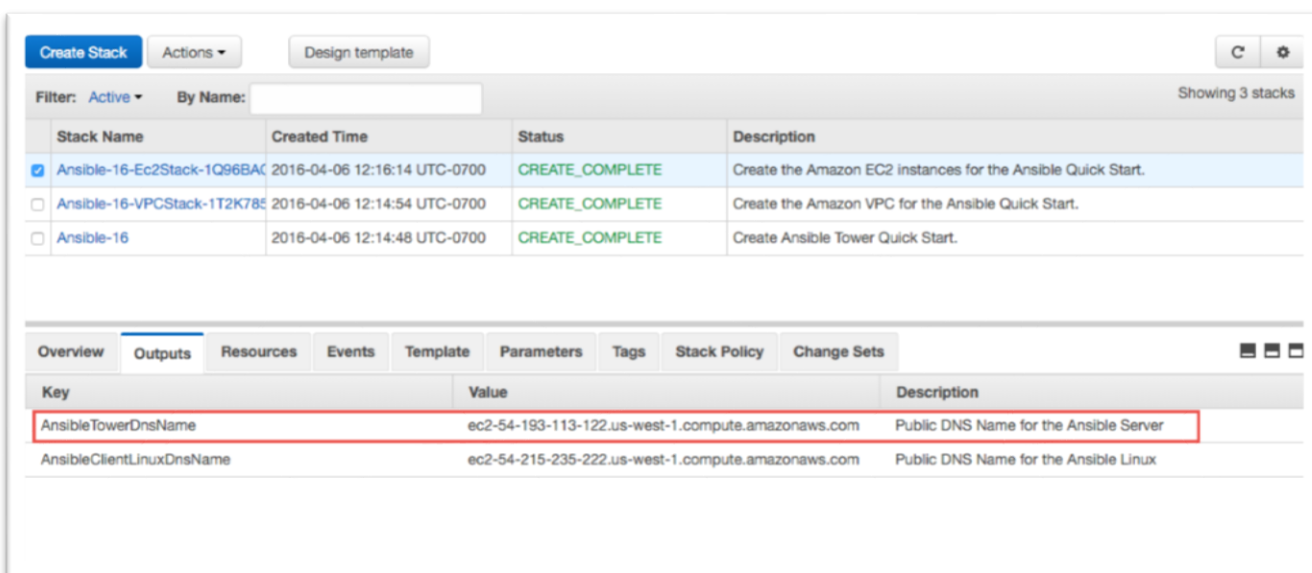
By default, this user has an IAM PowerUserAccess policy. If you wish, you can lock it down further.

For more information about IAM roles, see [IAM Roles for Amazon EC2](#) in the Amazon EC2 documentation.

Step 4. Get a Trial License for Ansible Tower

1. Navigate to the IP or host name of Ansible Tower in your Amazon VPC.

Note You can find the IP address for Ansible Tower on the **Outputs** tab of the AWS Management Console, as shown in Figure 8, or in the Amazon EC2 console.



Stack Name	Created Time	Status	Description
<input checked="" type="checkbox"/> Ansible-16-Ec2Stack-1Q96BAK	2016-04-06 12:16:14 UTC-0700	CREATE_COMPLETE	Create the Amazon EC2 instances for the Ansible Quick Start.
<input type="checkbox"/> Ansible-16-VPCStack-1T2K785	2016-04-06 12:14:54 UTC-0700	CREATE_COMPLETE	Create the Amazon VPC for the Ansible Quick Start.
<input type="checkbox"/> Ansible-16	2016-04-06 12:14:48 UTC-0700	CREATE_COMPLETE	Create Ansible Tower Quick Start.

Key	Value	Description
AnsibleTowerDnsName	ec2-54-193-113-122.us-west-1.compute.amazonaws.com	Public DNS Name for the Ansible Server
AnsibleClientLinuxDnsName	ec2-54-215-235-222.us-west-1.compute.amazonaws.com	Public DNS Name for the Ansible Linux

Figure 8: IP Address for Ansible Tower

You'll see a warning in your web browser. This is because the deployment uses a self-signed certificate. To proceed, you'll need to trust the self-signed certificate, but we recommend that you add your own certificate to Ansible Tower for higher security. For more information, see the [Ansible Tower User Guide](#).

2. Choose **Advanced**, and then choose **Proceed** to reach the Ansible Tower dashboard. (This instruction reflects Google Chrome settings. The options for proceeding will depend on your web browser.)
3. For the user name, type **admin**, and then provide the administrator password you specified for the **AnsibleAdminPassword** parameter when you launched the Quick Start in [step 2](#).

The image shows the Ansible Tower sign-in interface. At the top, there is a black banner with the text 'ANSIBLE TOWER' in white and teal. Below the banner, a blue message says 'Welcome to Ansible Tower! Please sign in.' There are two input fields: one for 'Username' with the text 'admin' and one for 'Password' with masked characters '.....'. A blue 'Sign in' button with a right-pointing arrow is located at the bottom right of the form.

Figure 9: Ansible Sign-in Screen

If you forgot your password, you'll find it in the `tower_setup_conf.yml` file in your root home directory. The following command will display the administrator password:

```
[centos@ip-10-0-0-10 ~]$ sudo grep admin_password /root/ansible-tower-setup-bundle-2.4.4-1.el7/tower_setup_conf.yml
```

Warning The `tower_setup_conf.yml` file contains admin user credentials. We recommend that you delete this file after you retrieve your credentials.

If you need instructions on how to use Secure Shell (SSH) to connect into your EC2 instance, see [step 5](#).

Note On CentOS, use the user name `centos`, not `ec2-user`, for the SSH connection. For example, on CentOS, use:

```
ssh -i <your>.pem centos@<ip-or-hostname>
```


On RHEL, use:

```
ssh -i <your>.pem ec2-user@<ip-or-hostname>
```

4. Upon successful login, you will reach the screen shown in Figure 10. Click the red box to get a free Tower trial license. (If you already have a Tower license, you can skip this step and enter the license information, as shown in Figure 11.)
5. Choose the free trial option, enter the information requested, and then choose **Submit**.

SELECT TOWER EDITION

☐ **ENTERPRISE TOWER - FREE TRIAL**

For Enterprise environments that want to evaluate more than 10 nodes

Enterprise Tower includes powerful features not available in the basic edition:

- LDAP and Active Directory Support
- System tracking
- Audit trails
- Push-button self-service with surveys
- 8x5 or 24x7 phone & email support

☒ **10-NODE BASIC TOWER - FREE TRIAL**

Limited feature edition with a trial license that will not expire.
Support not included.

First Name*

Last Name*

Company Name*

Role*

Email*

Phone Number*

SUBMIT

Figure 10: Selecting the Free Trial for Ansible Tower

Ansible will send a trial license file to your email account.

[illegible]

Figure 11: Ansible Trial License File

6. Copy and paste the trial license file into the Ansible Tower dashboard.
7. Select the check box at the bottom of the screen if you agree with the End User License Agreement, and then choose **Submit**.

Step 5. Configure and Manage EC2 Instances in Tower

In this step, we'll configure Ansible Tower with Amazon EC2 integration, so we can view and manage EC2 instances in the Ansible Tower dashboard. We'll then add other managed instances for Ansible Tower to discover. For complete information about how you can set up organizations, teams, and projects in Ansible Tower, see the [Ansible Tower User Guide](#).

Configuring Ansible Tower with EC2 Integration

1. On the Ansible Tower dashboard, choose the button in the upper-right corner to open the setup screen.

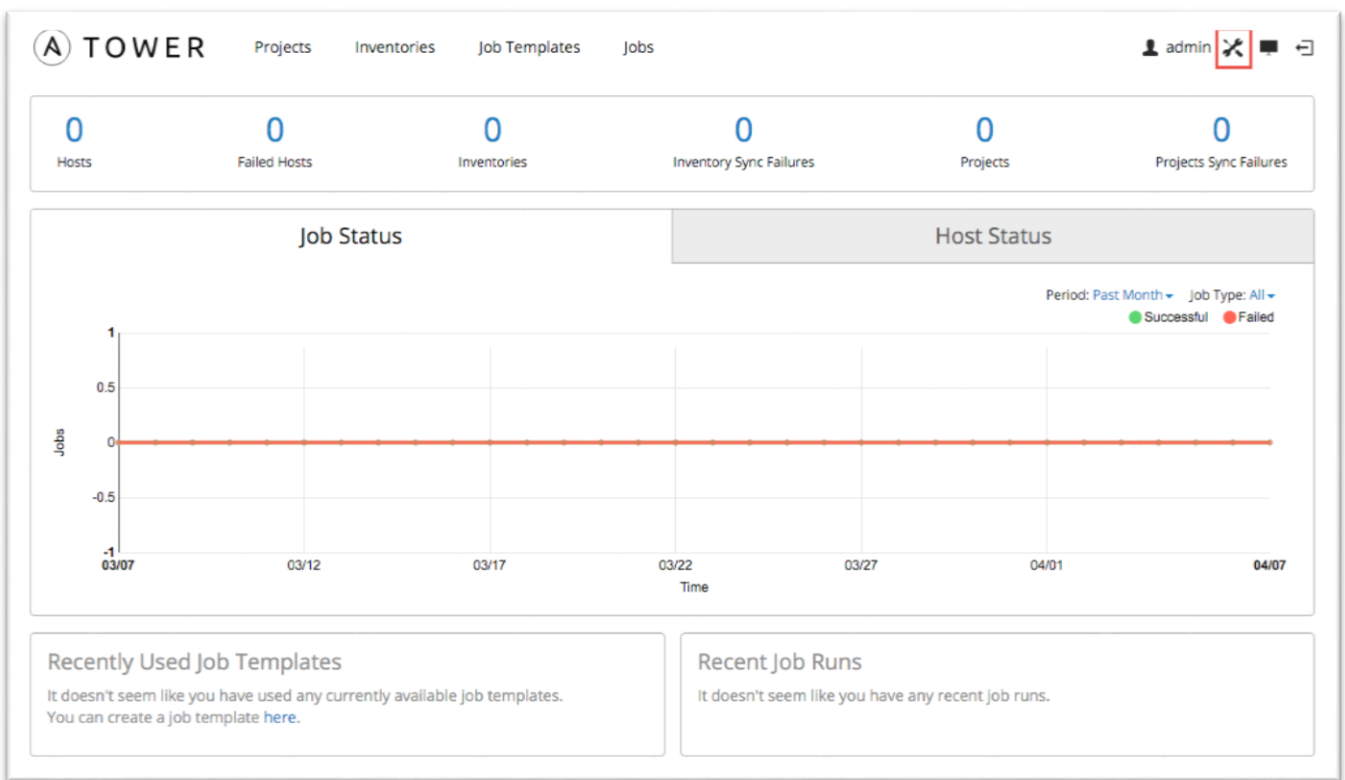



Figure 12: Ansible Tower Dashboard

2. Choose **Credentials**, and then choose the  button to create a new credential.
3. Enter the credential details as shown in Figure 13:
 - a. Provide a **Name** and **Description**.
 - b. Leave **User** selected as the owner type.
 - c. Select **admin** as the credential owner.
 - d. Choose **Amazon Web Services** as the credential type.
 - e. Enter your **Access Key** and **Secret Key**.
 - f. Choose **Save**.

The screenshot shows the 'Create Credential' form in the Ansible Tower web interface. The top navigation bar includes 'Projects', 'Inventories', 'Job Templates', and 'Jobs'. The user is logged in as 'admin'. The breadcrumb trail is 'Setup > Credentials > Create Credential'. The form contains the following fields:


- Name:** A text input field containing 'AWS Credentials'.
- Description:** A text input field containing 'ansibleadm'.
- Does this credential belong to a team or user?:** Radio buttons for 'User' (selected) and 'Team'.
- User that owns this credential:** A search input field containing 'admin'.
- Type:** A dropdown menu showing 'Amazon Web Services'.
- Access Key:** A text input field containing a masked string of 20 'X' characters.
- Secret Key:** A text input field with a 'Show' button and a masked string of 20 dots.
- STS Token:** A text input field with a 'Show' button and an empty field.

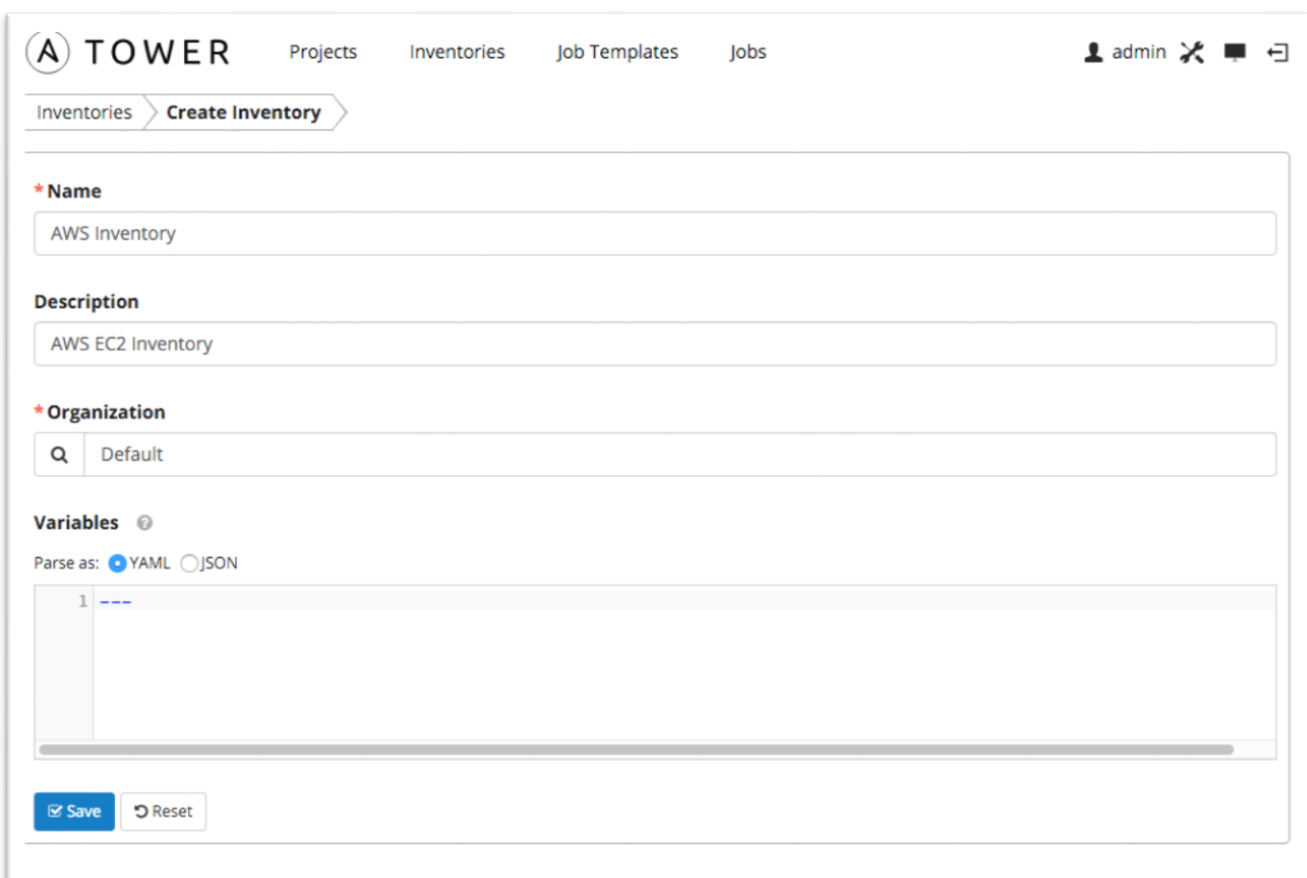
At the bottom of the form are two buttons: 'Save' (with a checkmark icon) and 'Reset' (with a circular arrow icon).

Figure 13: Creating a Credential in Ansible Tower

Discovering and Managing EC2 Instances in Ansible Tower

In Ansible, a collection of hosts that you can launch tasks against is called an *inventory*. Inventories consist of *groups*, which contain specific *hosts* (systems, nodes, or instances). For detailed information about these concepts, see the [Ansible Tower User Guide](#).

1. Create an inventory:
 - a. On the Ansible Tower dashboard, choose **Inventories**.
 - b. Choose the  button to open the **Create Inventory** window.
 - c. Provide a name and description for the default organization. Leave the **Variables** section unchanged, and choose **Save** to create an inventory.



The screenshot shows the 'Create Inventory' form in the Ansible Tower web interface. The top navigation bar includes 'TOWER', 'Projects', 'Inventories', 'Job Templates', and 'Jobs'. The user is logged in as 'admin'. The breadcrumb trail shows 'Inventories' > 'Create Inventory'. The form contains the following fields:

- * Name:** A text input field containing 'AWS Inventory'.
- Description:** A text input field containing 'AWS EC2 Inventory'.
- * Organization:** A dropdown menu with a search icon and the text 'Default'.
- Variables:** A section with a toggle for 'Parse as:' set to 'YAML' (radio button selected) and 'JSON' (radio button unselected). Below this is a large text area for variables, currently empty.

At the bottom of the form are two buttons: 'Save' (with a checkmark icon) and 'Reset' (with a circular arrow icon).

Figure 14: Creating an Inventory

2. Next, create a group:

- a. Choose the  button in the **Groups** section.

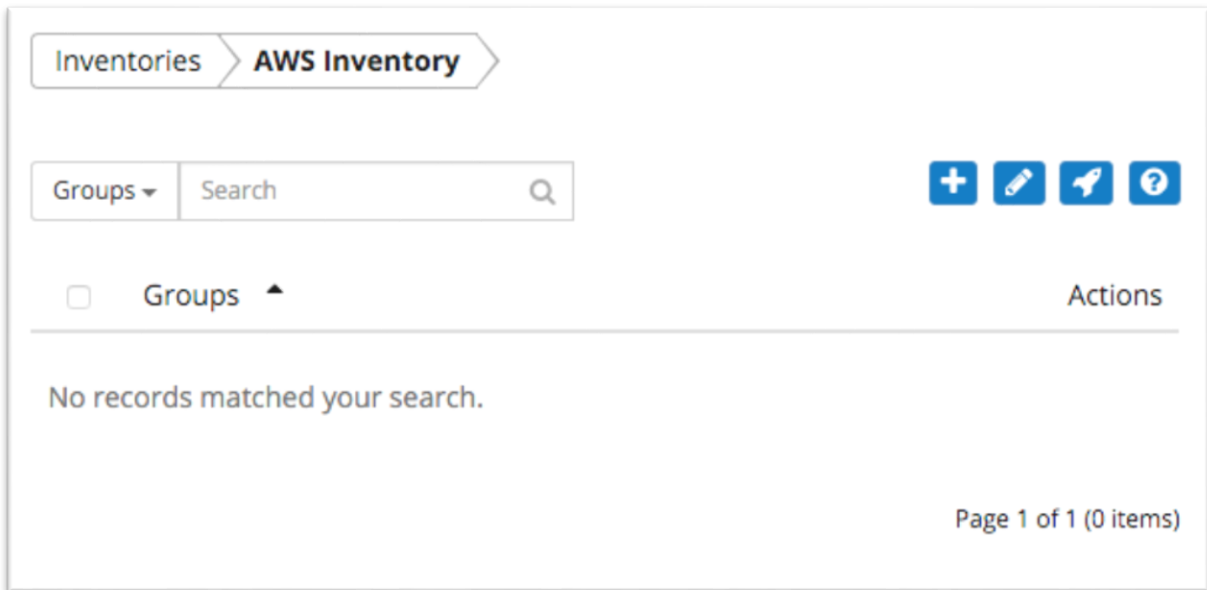


Figure 15: Groups in Ansible Tower

- b. In the **Add Group** window, **Properties** tab, provide a name and description for the group, as shown in Figure 16. Leave the **Variables** section unchanged, and then choose **Save**.

Add Group ✕

Properties

Source

*

Name

Ec2 Group

Description

Instances Discovery by EC2 Tag

Variables ?

Parse as: ☒ YAML ☐ JSON

1

✕ Cancel

✓ Save

Figure 16: Adding a Group for Amazon EC2

3. In the **Add Group** window, choose the **Source** tab.
 - a. From the **Source** list, choose **Amazon EC2**.
 - b. For **Cloud Credential**, choose the magnifying glass, and then choose **AWS Credentials**.
 - c. For **Region**, choose **All**.
 - d. For **Instance Filter**, type **tag-key=Tower**.
 - e. Leave **Only Group By tag-key=Tower** blank.
 - f. Leave the **Source Variables** and **Update Options fields** unchanged.
 - g. Choose **Save**.

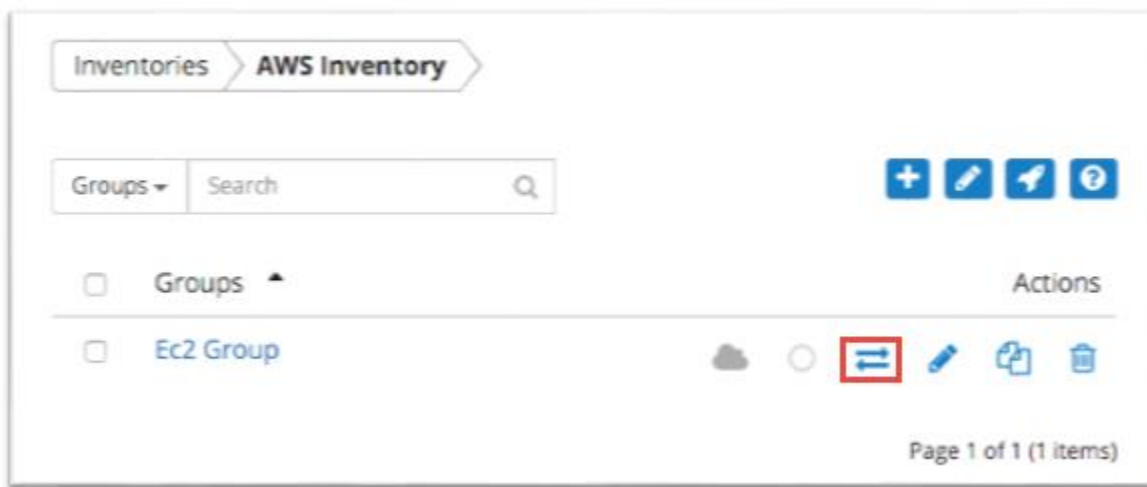


Figure 17: Your Amazon EC2 Group in the Ansible Tower Dashboard

4. In the **Groups** window, choose the  button for the Amazon EC2 group you created to start the synchronization process.

During synchronization, Ansible Tower will use the credentials you provided to pull metadata from AWS. When this process completes successfully, you will see a green cloud and a green circle, as shown in Figure 18, indicating that the synchronization with AWS is complete.

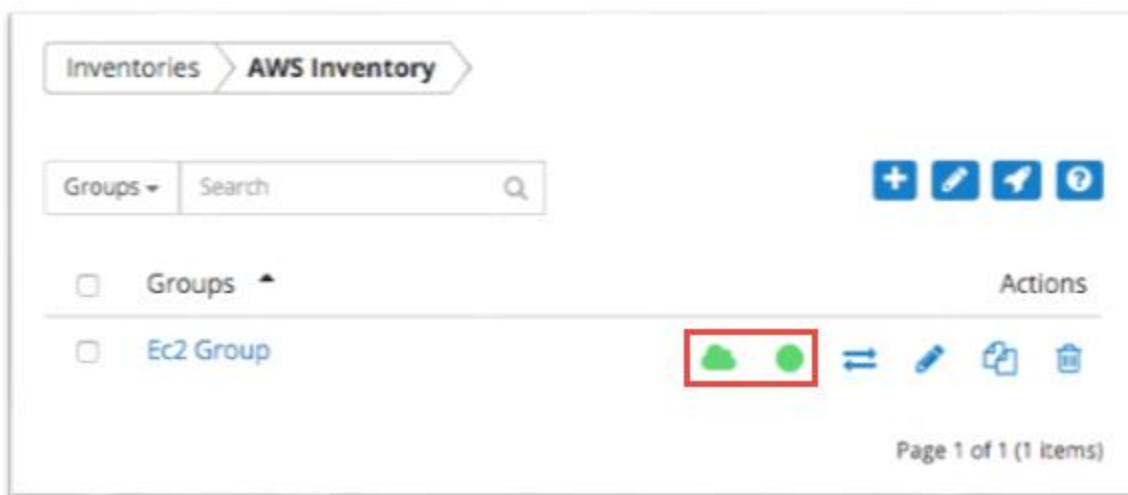


Figure 18: Synchronizing with AWS

In the **Hosts** section you will see two hosts: Ansible Tower and a bare Linux client instance, both managed by Ansible.

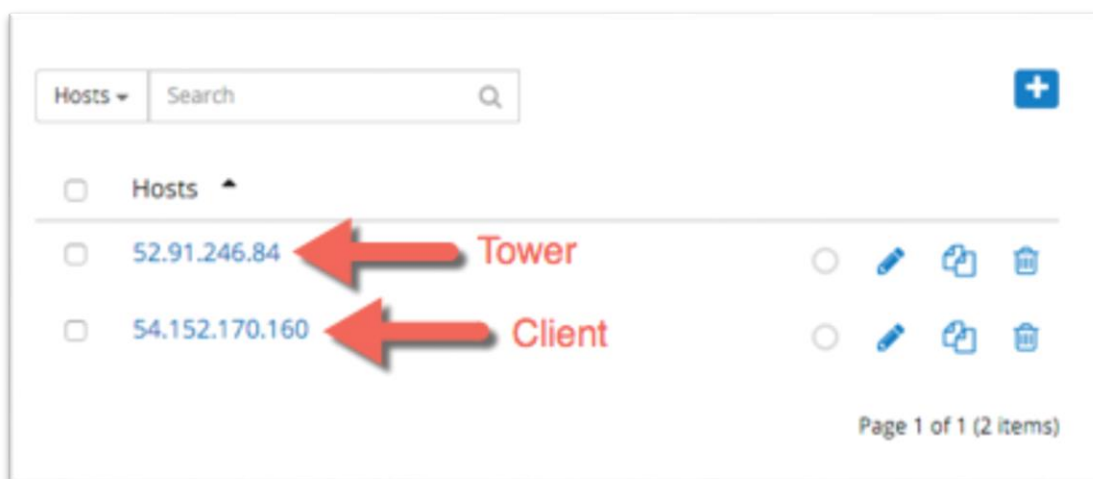


Figure 19: EC2 Instances Detected by Ansible Tower

5. You can choose the Tower host to see its instance metadata, as shown in Figure 20.

Host Properties

* Host Name ?

ansibletower

Description

imported

☒ Enabled? ?

Variables ?

Parse as: ☒ YAML ☐ JSON

```
1 ec2_in_monitoring_element: false
2 ec2_ami_launch_index: "0"
3 ec2_architecture: x86_64
4 ec2_client_token: Ansib-Ansib-100HZJ6BQLVBV
5 ec2_dns_name: ec2-52-91-246-84.compute-1.amazonaws.com
6 ec2_ebs_optimized: false
7 ec2_eventsSet: ""
8 ec2_group_name: ""
9 ec2_hypervisor: xen
10 ec2_id: i-0766449ce611e5d6b
11 ec2_image_id: ami-6d1c2007
12 ec2_instance_profile: ""
13 ec2_instance_type: t2.medium
14 ec2_ip_address: 52.91.246.84
15 ec2_item: ""
16 ec2_kernel: ""
17 ec2_key name: tonymv
18
```

✕ Cancel

✓ Save

Figure 20: Properties for the Ansible Tower Instance

Adding Other Managed Instances

The AWS CloudFormation template launched with this Quick Start initially created the EC2 instance and tagged it with the key **Tower**. The Amazon EC2 console provides a **Launch More Like This** feature that enables you to launch additional instances that will also be tagged with the key **Tower**. Ansible Tower will then discover the new instance when you request synchronization.

1. In the Amazon EC2 console, **Instances** page, select **AnsibleClient**.
2. Choose **Actions**, and then choose **Launch More Like This**.

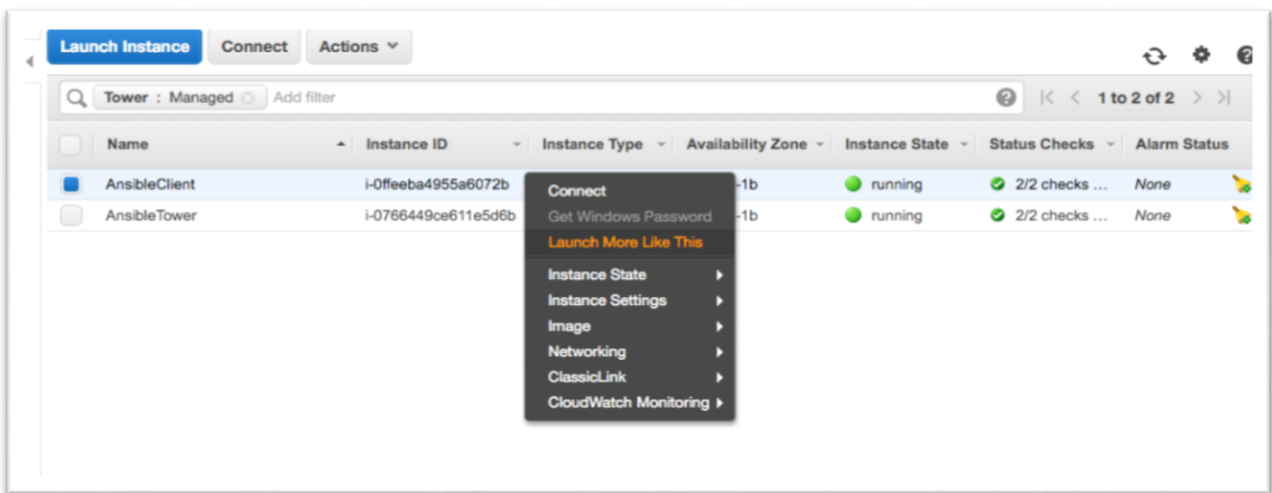


Figure 21: Launching Additional Amazon EC2 Instances

3. In Ansible Tower, choose the  button to scan for new instances.

In Figure 22, you can also see the third host. (To identify the Tower host more easily, I gave it a friendly name by updating the **Host Name** field shown in Figure 20.)

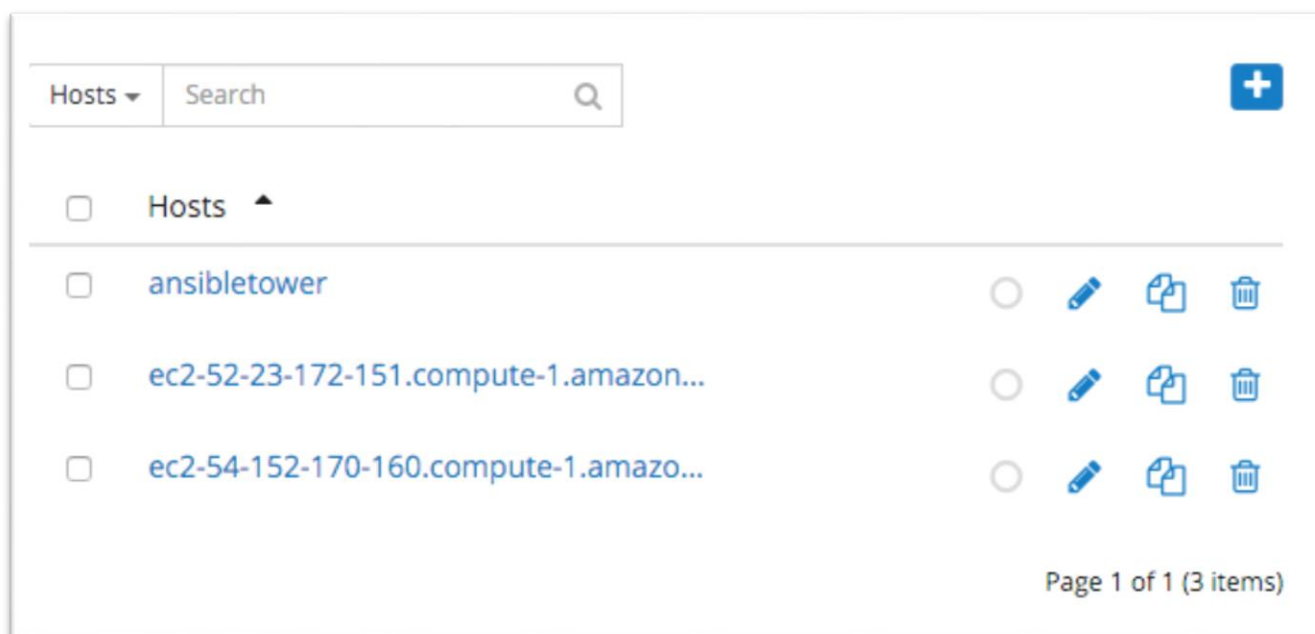


Figure 22: Discovering New EC2 Instances in Ansible Tower

Now that you've configured Ansible Tower to detect hosts that are tagged with the key **Tower**, you'll be able to deploy playbooks to manage and configure your EC2 instances.

Troubleshooting

When you deploy the Quick Start, if you encounter a **CREATE_FAILED** error instead of the **CREATE_COMPLETE** status code, we recommend that you relaunch the template with **Rollback on failure** set to **No**. (This setting is under **Advanced** in the AWS CloudFormation console, **Options** page.) With this setting, the stack's state will be retained and the instance will be left running, so you can troubleshoot the issue. (You'll want to look at the log files in %ProgramFiles%\Amazon\EC2ConfigService and C:\cfn\log.)

Important When you set **Rollback on failure** to **No**, you'll continue to incur AWS charges for this stack. Please make sure to delete the stack when you've finished troubleshooting.

For additional information, see [Troubleshooting AWS CloudFormation](#) on the AWS website.

Security

A *security group* acts as a firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time. The new rules are automatically applied to all instances that are associated with the security group.

The security groups created and assigned to the individual instances as part of this solution are restricted as much as possible while allowing access to the various functions needed by Ansible Tower. We recommend that you review security groups and further restrict access as needed once Ansible is up and running.

We highly recommend that you update the passwords for the administrator and database administrator accounts in accordance with your IT standards after you deploy the Quick Start.

Additional Resources

AWS services

- AWS CloudFormation
<http://aws.amazon.com/documentation/cloudformation/>
- Amazon EC2
<http://aws.amazon.com/documentation/ec2/>
- IAM
<http://aws.amazon.com/documentation/iam/>
- AWS OpsWorks
<http://aws.amazon.com/documentation/opsworks/>
- Amazon VPC
<http://aws.amazon.com/documentation/vpc/>

Ansible resources

- Ansible Tower User Guide
<http://docs.ansible.com/ansible-tower/2.2.0/html/userguide/>

Quick Start Reference Deployments

- AWS Quick Start home page
<https://aws.amazon.com/quickstart/>
- Quick Start deployment guides
<https://aws.amazon.com/documentation/quickstart/>

Send Us Feedback

We welcome your questions and comments. Please post your feedback on the [AWS Quick Start Discussion Forum](#).

Document Revisions

Date	Change	In sections
April 2016	Initial publication	-