Ansible Tower on the AWS Cloud

Quick Start Reference Deployment

Tony Vattathil  
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](#_Toc448217033)

[Ansible Tower on AWS 3](#_Toc448217034)

[Quick Links 4](#_Toc448217035)

[Cost and Licenses 4](#_Toc448217036)

[Ansible Tower Licenses 4](#_Toc448217037)

[Ansible Licenses 4](#_Toc448217038)

[Architecture 5](#_Toc448217039)

[AWS Services 6](#_Toc448217041)

[Ansible Tower Installation 6](#_Toc448217042)

[Automated Deployment 7](#_Toc448217043)

[What We’ll Cover 7](#_Toc448217044)

[Step 1. Prepare an AWS Account 8](#_Toc448217045)

[Step 2. Launch the Ansible Tower Stack 11](#_Toc448217046)

[Step 3. Create a User Account for Ansible Tower 13](#_Toc448217047)

[Step 4. Get a Trial License for Ansible Tower 14](#_Toc448217048)

[Step 5. Configure Ansible Tower with EC2 Integration 19](#_Toc448217049)

[Step 6. Discover and Manage EC2 Instances in Ansible Tower 20](#_Toc448217050)

[Step 7. Add Other Managed Instances 26](#_Toc448217051)

[Troubleshooting 28](#_Toc448217052)

[Security 29](#_Toc448217053)

[Additional Resources 29](#_Toc448217054)

[Send Us Feedback 30](#_Toc448217055)

[Document Revisions 30](#_Toc448217056)

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 also provides links for viewing and launching the [AWS CloudFormation](http://aws.amazon.com/cloudformation/) template that automates the Ansible Tower deployment and creates Ansible-aware Amazon Elastic Compute Cloud (Amazon Ec2) instances. The guide also details 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.

[Quick Starts](http://aws.amazon.com/quickstart/) 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.

# Overview

## Ansible Tower on AWS

Ansible is an IT DevOps tool that automates cloud 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.

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 resources for your system. Ansible executes these modules over Secure Shell (SSH) by default, and removes them when finished.

## Quick Links

**Launch   
Quick Start**

If you have an AWS account and you’re already familiar with AWS and Ansible Tower, you can [launch the Quick Start](https://console.aws.amazon.com/cloudformation/home?region=us-west-2#/stacks/new?stackName=AnsibleTower&templateURL=https:%2F%2Fs3.amazonaws.com%2Fquickstart-reference%2Fansible%2Flatest%2Ftemplates%2FAnsible-Tower-QuickStart.template) to build the architecture shown in [Figure 1](#_Architecture_Overview). 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](#_Automated_Deployment_and) provided later in this guide   
to launch the Quick Start.

**View template**

If you want to take a look under the covers, you can [view the   
AWS CloudFormation template](https://s3.amazonaws.com/quickstart-reference/ansible/latest/templates/Ansible-Tower-QuickStart.template) that automates this deployment. The default configuration deploys three servers that use the t2.medium instance type by default.

## 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. As of the date of publication, the cost for using the Quick Start with default settings is approximately $0.18 an hour. Prices are subject to change. See the pricing pages for each AWS service you will be using or the [AWS Simple Monthly Calculator](http://calculator.s3.amazonaws.com/index.html) for full details.

This Quick Start deploys Ansible Tower on the AWS cloud with Linux Ansible client. Ansible and Ansible Tower have separate licensing models.

### Ansible Tower Licenses

Ansible Tower is subject to the [Ansible Software Subscription and Services Agreement](http://www.ansible.com/subscription-agreement). Ansible Tower is a proprietary product offered by Ansible, Inc. and its use is not intended to prohibit the rights under any open source license.

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.

### Ansible Licenses

Ansible is licensed under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License).

# Architecture

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

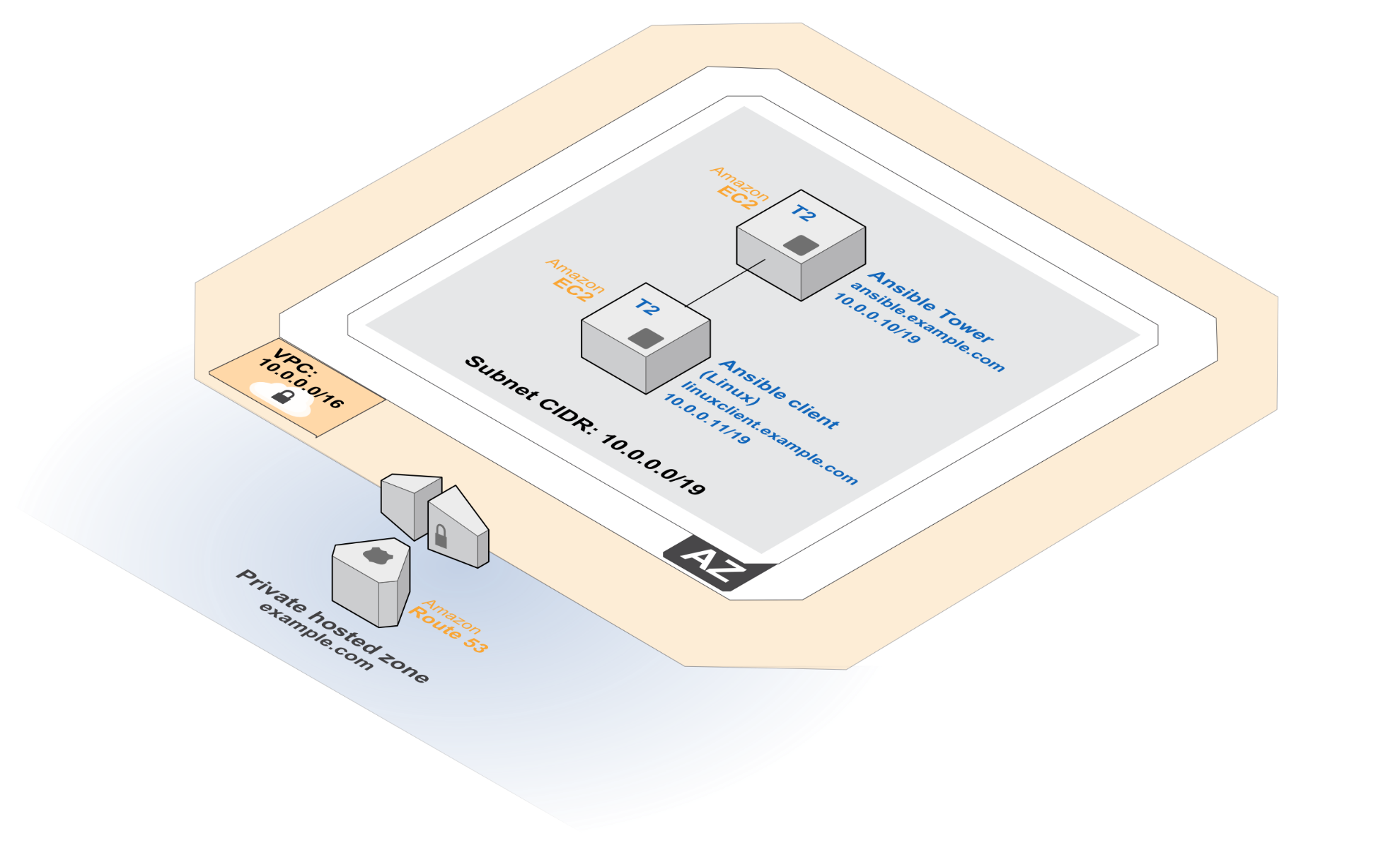


Figure 1: Quick Start Architecture for Ansible Tower on AWS

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 a random Availability Zone.
* One Linux client instance is deployed into the VPC subnet. This Quick Start uses CentOS version 7 as the Linux client, but you can attach a Red Hat Enterprise Linux (RHEL) client instead by using the same procedure detailed in this document.
* 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](http://docs.aws.amazon.com/gettingstarted/latest/awsgsg-intro/intro.html) of the AWS documentation.)

* [Amazon VPC](http://aws.amazon.com/documentation/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.
* [Amazon EC2](http://aws.amazon.com/documentation/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.

## Ansible Tower Installation

This Quick Start deploys Ansible Tower on an EC2 instance that is running CentOS version 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 the 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 6 of this guide.

# Automated Deployment

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 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 3. 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. Get Ansible Tower License

* Connect to your Ansible Tower via a web browser and follow the steps to license Ansible Tower.
* Follow the walkthrough to integrate AWS compute resources into the Tower inventory.

## 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*](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html). 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.

1. Create a [key pair](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html) 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**.

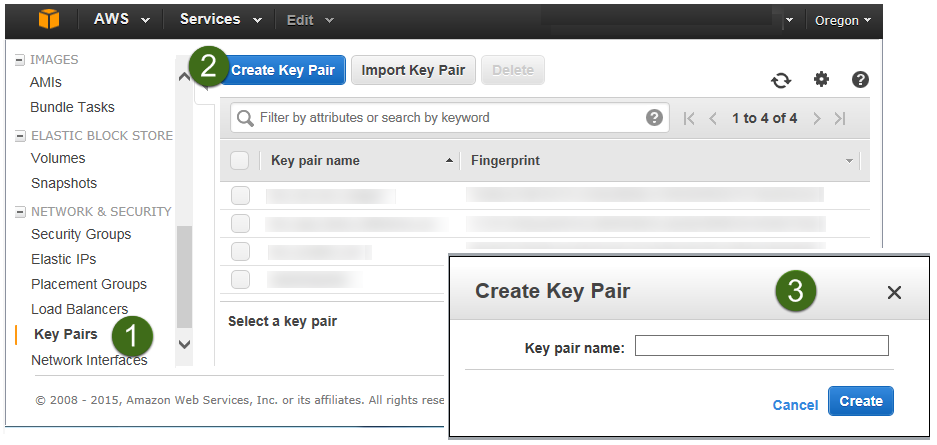


Figure 3: Creating a Key Pair

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.

1. If you would like to use a different instance type, [request a service limit increase](https://console.aws.amazon.com/support/home#/case/create?issueType=service-limit-increase&limitType=service-code-) 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 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](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-resource-limits.html) in the AWS documentation.

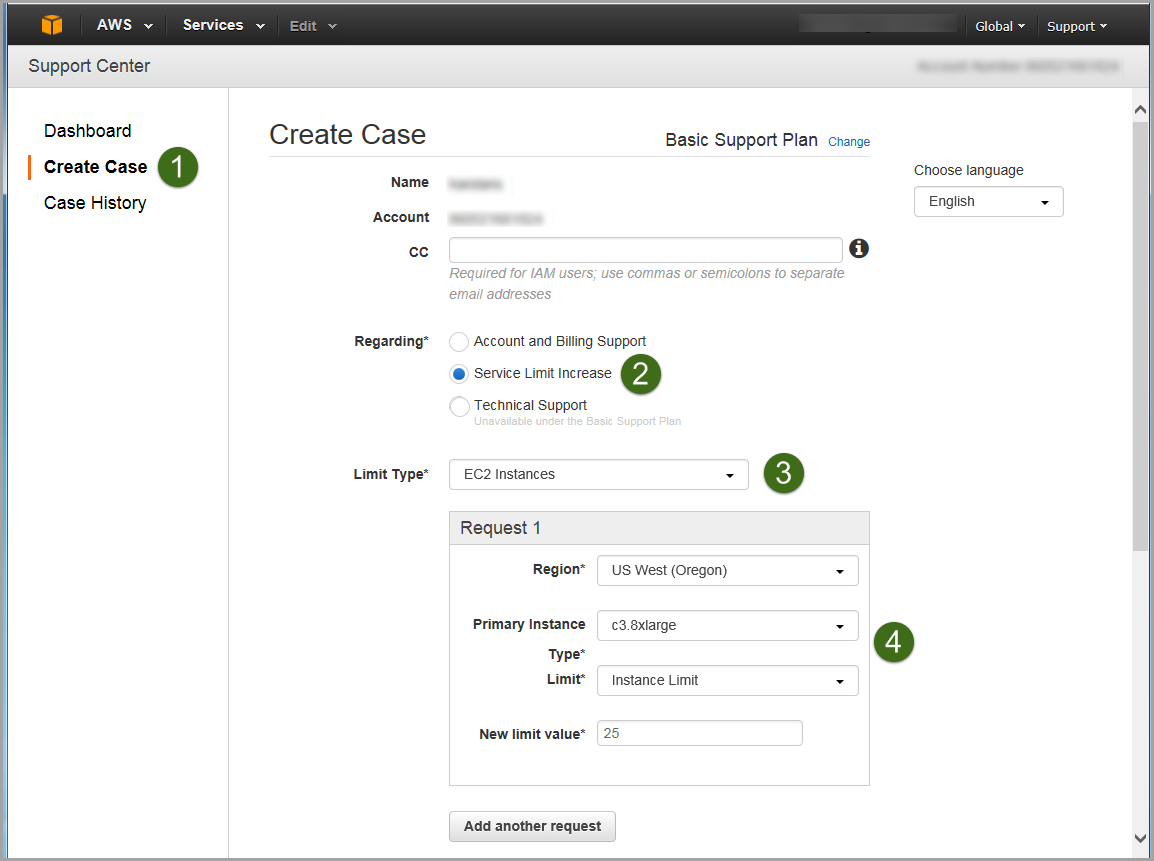


Figure 4: Requesting a Service Limit Increase

## 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.

**Launch   
Quick Start**

1. Launch the AWS CloudFormation template into your AWS account.

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.

**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 template](https://s3.amazonaws.com/quickstart-reference/ansible/latest/templates/Ansible-Tower-QuickStart.template) to use it as a starting point for your own implementation.

1. On the **Select Template** page, keep the default URL for the AWS CloudFormation template, and then choose **Next**.
2. On the **Specify Details** page, review the parameters for the template. These are described in the following table. Provide values for the **KeyPairName**, **AnsibleAdminPassword**, and **DatabaseAdminPassword** 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 label | Parameter name | Default | Description |
| Select a key pair | **KeyPairName** | *Requires input* | 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. |
| Set you Tower Admin Password | **AnsibleAdminPassword** | *Requires input* | 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$ibl3ChgMe). |
| Specify a DataBaseAdminPassword | **DatabaseAdminPassword** | *Requires input* | Password for the Ansible database administrator account. All database resources (e.g., Redis, Postgres) will use this password. |
| Source IP for remote access | **RemoteAdminCIDR** | 0.0.0.0/0 | CIDR block or IP address for SSH (e.g., 1.1.1.1/32). |
| CIDR range for your VPC | **VPCCIDR** | 10.0.0.0/16 | CIDR block for the Amazon VPC. |
| CIDR range for the subnet in your VPC | **SubnetCIDR** | 10.0.0.0/19 | CIDR block for the public subnet in the Amazon VPC where Ansible Tower will be deployed. |
| IP address for the Ansible Tower | **AnsibleTowerIP** | 10.0.0.10 | Private IP of your Ansible Tower. |
| IP address for the Linux Client. (bare Linux instance) | **AnsibleClientLinuxIP** | 10.0.0.11 | IP address of a Linux instance that can be managed by Ansible. |

**Note** You can also [download the template](https://s3.amazonaws.com/quickstart-reference/ansible/latest/templates/Ansible-Tower-QuickStart.template) and edit it to create your own parameters based on your specific deployment scenario.

1. On the **Options** page, you can [specify tags](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-resource-tags.html) (key-value pairs) for resources in your stack and [set additional options](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-console-add-tags.html). When you’re done, choose **Next**.
2. On the **Review** page, review and confirm the settings.
3. Choose **Create** to deploy the stack.
4. Monitor the status of the stack. When the status is **CREATE\_COMPLETE**, Ansible Tower is ready to use. The IP address is displayed in the **Outputs** tab for the stack, as shown later in Figure 8.

## Step 3. Create a User Account for Ansible Tower

**As part of this setup I created a new user called *ansibleadm*. This is an optional step. You can use an existing user account instead, if that account has the proper privileges.**

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.**

****

Figure 5: Creating a User Account for Ansible Tower

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

****

Figure 6: Security Credentials for User Account

1. 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

I gave this user PowerUser IAM privileges. If you wish, you can lock it down further.

For more information about IAM roles, see [IAM Roles for Amazon EC2](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html) 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 Ec2Stack or in the Amazon Ec2 console.

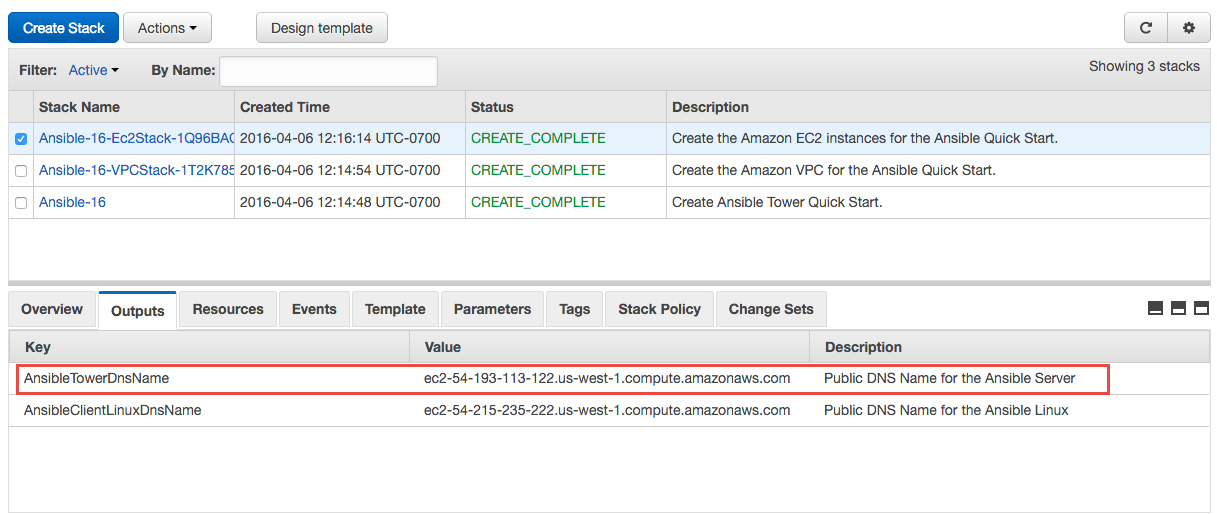


Figure 8: IP Address for Ansible Tower

You’ll see a warning in your web browser similar to Figure 9. This is because the deployment uses a self-signed certificate. To proceed, you will need to trust the self-signed certificate, but we recommend that you add your own certificate to Ansible Tower for higher security. See the user guide in the appendix for more info.

**Note** The warning and options you’ll see depend on your web browser. Figures 9 and 10 show the warning that’s displayed in Google Chrome.



Figure 9: Self-Signed Certificate Warning

1. Choose **Advanced**, and then choose **Proceed** to reach the Ansible Tower dashboard.



Figure 10: Accepting the Self-Signed Certificate

The Ansible Tower sign-in screen is displayed with **admin** as the default user name.

1. Provide the administrator password you specified for the **AnsibleAdminPassword** parameter when you launched the Quick Start in [step 2](#_Step_2._Launch).



Figure 11: 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 ssh into your ec2-instance, please see the section on accessing ec2 instances

Note: On CentOS the username to connect vi ssh is centos not ec2-user

example command(CentOS): ssh –i your.pem centos@ip-or-hostname

example command(RHEL): ssh –i your.pem centos@ip-or-hostname

1. Upon successful login, you will reach the screen show below. Click the red box to Get a Free Tower Trial License.
2. On the next screen illustrated in Figure 12, choose the free trial option,enter the information requested, and then choose **Submit**.



Figure 12: Selecting the Free Trial for Ansible Tower

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



Figure 13: Ansible Trial License File

1. Copy and paste the trial license file into the UI.
2. 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 Ansible Tower with EC2 Integration

In this step, we are going to configure Ansible Tower with Amazon EC2 integration, so we can view and manage Ec2 instances in the Ansible Tower dashboard. For complete information about how you can set up organizations, teams, and projects in Ansible Tower, see the [Ansible Tower User Guide](http://docs.ansible.com/ansible-tower/2.2.0/html/userguide/).

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

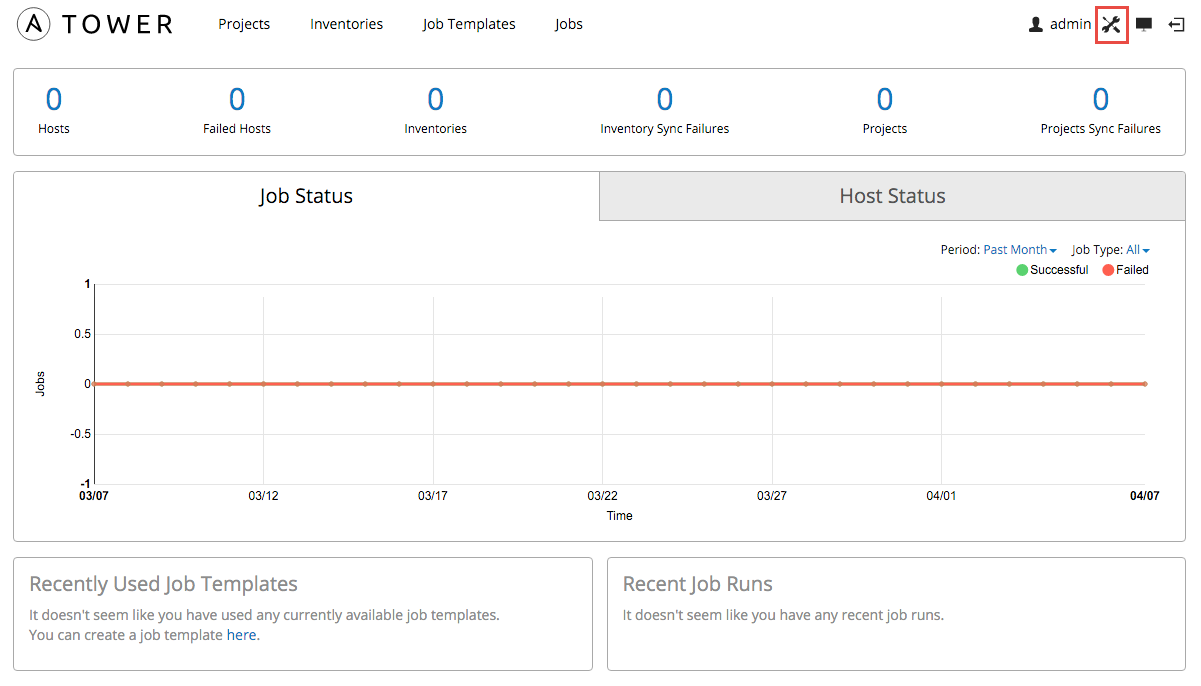


Figure 13: Ansible Tower Dashboard

1. Choose **Credentials**, and then choose the  button to create a new credential.
2. Enter the credential details, as shown in Figure 14:
3. Provide a **Name** and **Description**.
4. Leave **User** selected as the owner type.
5. Select **admin** as the credential owner.
6. Choose **Amazon Web Services** as the credential type.
7. Enter your **Access Key** and **Secret Key**.
8. Choose **Save**.



Figure 14: Creating a Credential in Ansible Tower

## Step 6. Discover and Manage EC2 Instances in Ansible Tower

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



Figure 15: Creating an Inventory

1. Next, create a group:
2. Choose the  button in the **Groups** section.



Figure 16: Groups in Ansible Tower

1. In the **Add Group** window, **Properties** tab, provide a name and description for the group, as shown in Figure 17. Leave the **Variables** section unchanged, and then choose **Save**.



Figure 17: Adding a Group for Amazon EC2

1. In the **Add Group** window, choose the **Source** tab.
2. From the **Source** list, choose **Amazon EC2**.
3. For **Cloud Credential**, choose the magnifying glass, and then choose **AWS Credentials.**
4. For **Region,** choose **All.**
5. For **Instance Filter,** type **tag-key=Tower**.
6. Leave **Only Group By tag-key=Tower** blank.
7. Leave the **Source Variables** and **Update Options fields** unchanged.
8. Choose **Save.**



Figure 18: Your Amazon EC2 Group in the Ansible Tower Dashboard

1. In the **Groups** window, choose the button for the Amazon EC2 group you created to start the synchronization process. When this process completes successfully, you will see a green cloud and a green circle, as shown in Figure 19, indicating that the synchronization with AWS is complete.



Figure 19: Synchronizing with AWS

In the **Hosts** section you will see two hosts (EC2 instances) associated with the group: Ansible Tower and a bare Linux instance, both managed by Ansible.



Figure 20: EC2 Instances Detected by Ansible Tower

1. You can click the host to see the discovery info.



Figure 21: Properties for the Ansible Tower Instance

## Step 7. Add 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 by using Tower as the instance template. 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 22: Launching Additional Amazon EC2 Instances

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

In Figure 23, you can also see the second host, which I’ve given a friendly name.



Figure 23: Discovering New EC2 Instances in Ansible Tower

# Troubleshooting

If you encounter a CREATE\_FAILED error, 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.

**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](http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/troubleshooting.html) on the AWS website. If the problem you encounter isn’t covered on that page above, please visit the [AWS Support Center](https://console.aws.amazon.com/support/).

# 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.

# Additional Resources

**AWS services**

* Amazon EC2  
  <http://aws.amazon.com/documentation/ec2/>
* AWS CloudFormation  
  <http://aws.amazon.com/documentation/cloudformation/>
* 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](https://forums.aws.amazon.com/forum.jspa?forumID=178).

# Document Revisions

|  |  |  |
| --- | --- | --- |
| Date | Change | In sections |
| April 2016 | Initial publication |  |