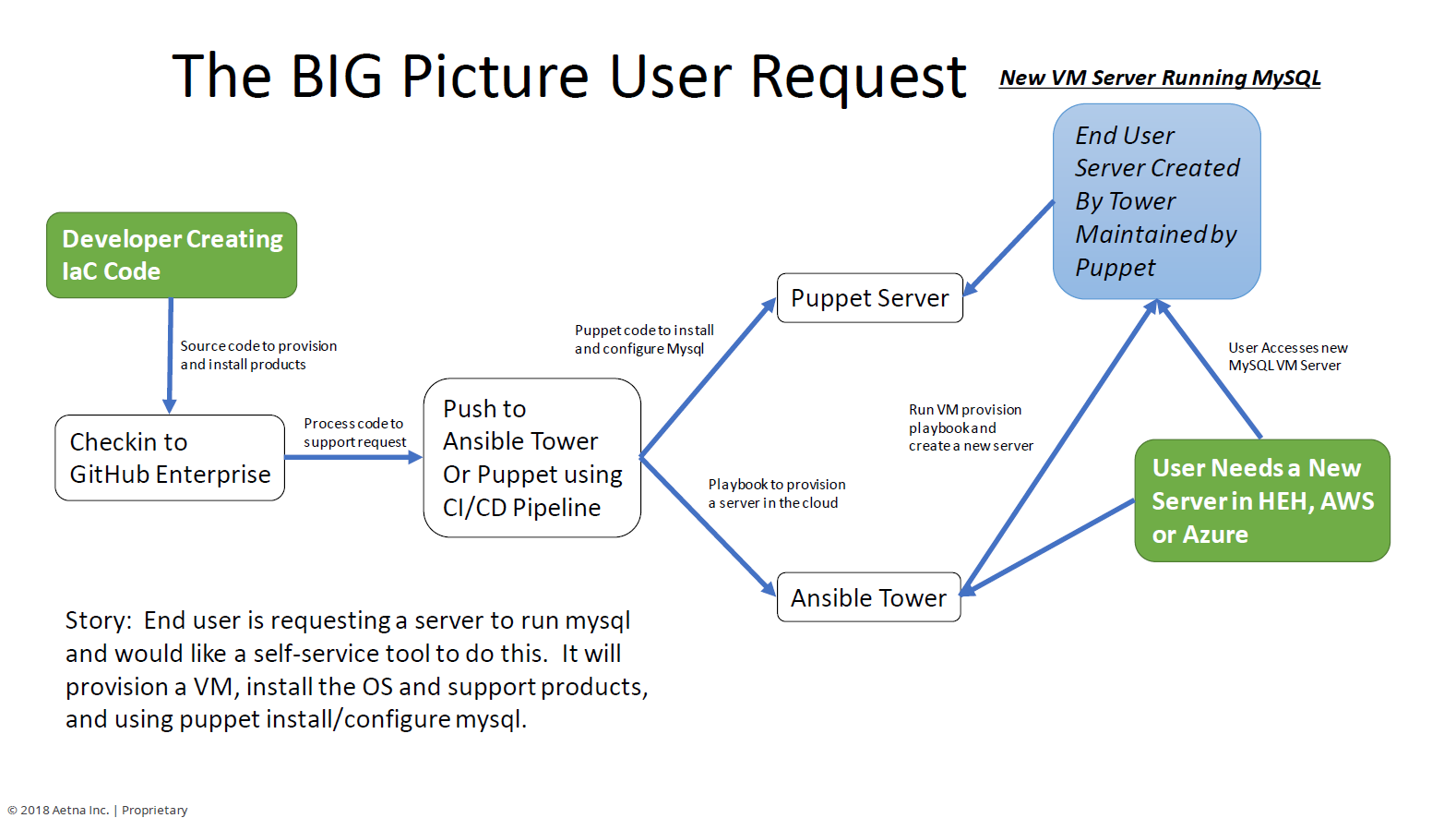
Getting Started

Introduction

Thank you for your interest in leveraging the Aetna Infrastructure as Code framework. This readme file is intended to provide you enough information to get your automation development journey started. This will cover the two pipelines that Aetna supports, and help guide you to the proper project setup. We need to establish some terminology first.



IaC Pipeline Guidance

At the date of this writing, Aetna support two separate CICD pipelines.

1. Core IaC CICD Pipeline
2. HealtheHost IaC CICD Pipeline

When you consider creating your automation project, you need to understand your target VM type in order to determine which pipeline you need to use. In some cases the answer will be BOTH pipelines.

Target VM Types

1. **Core VMs –** These exist in Aetna’s Data Centers, and in the Public Cloud (AWS, Azure, etc..)
2. **HealtheHost VMs –** These exist in Aetna on-prem Cloud (Openstack) and in the Public Cloud (AWS, Azure, etc..).

**Core VMs** are managed by Aetna internal endpoints (Ansible, Puppet) and attached to internal Active Directory domains for Identity & Access Management. (AETH & AETT)

**HealtheHost VMs** are managed by Aetna HealtheHost endpoints (Ansible, Puppet) and attached to HealtheHost Active Directory domains for Identity & Access Management. (HealtheHost and HealtheHostT)

If you are writing automation and understand your Target VMs based on the definitions above, use the following flowchart to guide you toward creating your projects in the proper CICD pipeline(s).

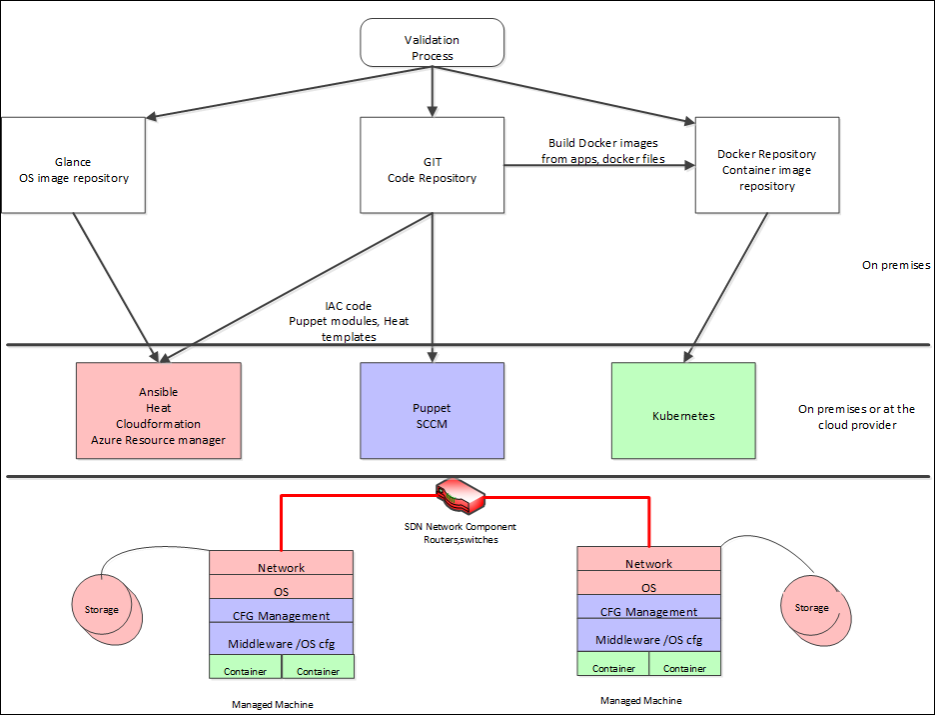
AeTH and HeH Pipelines

* You will need an account in the AeTH/AETT or HealtheHost/HealtheHostT
  + Instructions for setting up AETT accounts [AETT User Account Guide](https://github.aetna.com/pages/IaC/Docs/files/Setting_up_AETT_Accounts.pdf)
  + Instructions for setting up Healthehost account and groups [Tuebora Guide](https://github.aetna.com/pages/IaC/Docs/files/AccessNow_HOW_TO.pdf)
* Your project must be located in the AeTH and HeH IaC Pipeline
* AeTH Entry Point - [http://Github.Aetna.com](http://github.aetna.com/)
* HeH Entry Point - [http://Git.Aetna.com](http://git.aetna.com/)

Do I use Puppet or Ansible?

**Ansible = Do, Puppet = keep** Translation:

* If you need to perform a task (do) Ansible is really good at that and can perform and Orchestrate tasks well
* If you want to establish a configuration and manage its drift (keep) puppet really excels at that
* So (in my opinion) we should use Ansible to orchestrate the provisioning of a VM (Base VM from image, networking) and Puppet should maintain the configuration of that image going forward.
* As engineers we can get stuck in the religion of products easily. Honestly we could use either of these (or many other products) to do everything . But here is my analogy (which I always have an analogy!!) have you ever tried cutting a branch with the saw on the swiss army knife? It will work but a chainsaw is just made for that task



Puppet and Ansible

* [Puppet](https://github.aetna.com/pages/IaC/Docs/gettingstarted/?tabs=tabid-10%2Ctabid-1%2Ctabid-3%2Ctabid-5%2Ctabid-51#tabpanel_CeZOj-G++Q_tabid-10)

* [Ansible](https://github.aetna.com/pages/IaC/Docs/gettingstarted/?tabs=tabid-10%2Ctabid-1%2Ctabid-3%2Ctabid-5%2Ctabid-51#tabpanel_CeZOj-G++Q_tabid-20)

Puppet is a great choice to install, run, and maintain the state of your environment. The responsibility of coding the Puppet module(s) falls on the team who manages the environment.

**Puppet - Keeper** Puppet will be installed and configured in order to support product/software install, configuration and state/drift management. Some examples are:

* Postgresql
* MySQL
* Puppet Agent
* IIS
* Apache
* Tripwire, Splunk, etc
* Any other product that requires installation and state management

**For more information, visit the**[**Puppet page**](https://github.aetna.com/pages/IaC/Docs/iacservices/puppet/index.html)

**TIP**

For questions about Puppet please contact:

* **Peter Krawetzky (**[**IM**](sip:KrawetzkyPJ@aetna.com)**| [e-mail](**[**KrawetzkyPJ@aetna.com?subject=Puppet**](mailto:KrawetzkyPJ@aetna.com?subject=Puppet)**Questions) |**[**phone**](tel:860-273-0301)**)**
* **John Dimauro (**[**IM**](sip:DimauroJ@aetna.com)**| [e-mail](**[**DimauroJ@aetna.com?subject=Puppet**](mailto:DimauroJ@aetna.com?subject=Puppet)**Questions))**
* **Mark Stalpinski (**[**IM**](sip:StalpinskiM@aetna.com)**| [e-mail](**[**StalpinskiM@aetna.com?subject=Puppet**](mailto:StalpinskiM@aetna.com?subject=Puppet)**Questions) |**[**phone**](tel:860-273-1064)**)**
* **Andy Kim (**[**IM**](sip:aykim@aetna.com)**| [e-mail](**[**aykim@aetna.com?subject=Puppet**](mailto:aykim@aetna.com?subject=Puppet)**Questions) |**[**phone**](tel:602-659-1084)**)**
* **Steven Alexson (**[**IM**](sip:AlexsonS@aetna.com)**| [e-mail](**[**AlexsonS@aetna.com?subject=Puppet**](mailto:AlexsonS@aetna.com?subject=Puppet)**Questions) |**[**phone**](tel:860-273-5393)**)**

Prerequisites

* [AeTH](https://github.aetna.com/pages/IaC/Docs/gettingstarted/?tabs=tabid-10%2Ctabid-1%2Ctabid-3%2Ctabid-5%2Ctabid-51#tabpanel_CeZOj-G++Q-1_tabid-1)

* [HeH](https://github.aetna.com/pages/IaC/Docs/gettingstarted/?tabs=tabid-10%2Ctabid-1%2Ctabid-3%2Ctabid-5%2Ctabid-51#tabpanel_CeZOj-G++Q-1_tabid-2)

1x AD (Active Directory) Group

1x DL (Distribution List)

* Know the name of the puppet module or ansible project
* know the name of what organization they will use in Github
* Organizations are usually the names of the team you're working under

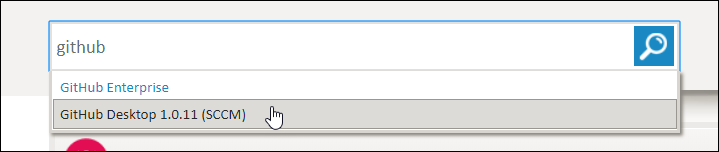
STEP 1: Get license for GHE

Since a license is required to access GitHub you must request Github in the app store.

1. Visit the Aetna [Appstore](https://appstore.aetna.com/) and request [GitHub Enterprise (GitHub)](https://appstore.aetna.com/Product/Details/74307). Once accepted you will be a member of Inst\_GitHubEnterprise group and your department will be charged $250.
2. Once you are approved, you can log into the [https://github.aetna.com](https://github.aetna.com/) URL to gain access.

STEP 2: Accessing GitHub

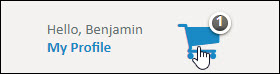
* Visit the Aetna [Appstore](https://appstore.aetna.com/) and, as of this writing April 2018, search for "GitHub Desktop 1.0.11 (SCCM)".



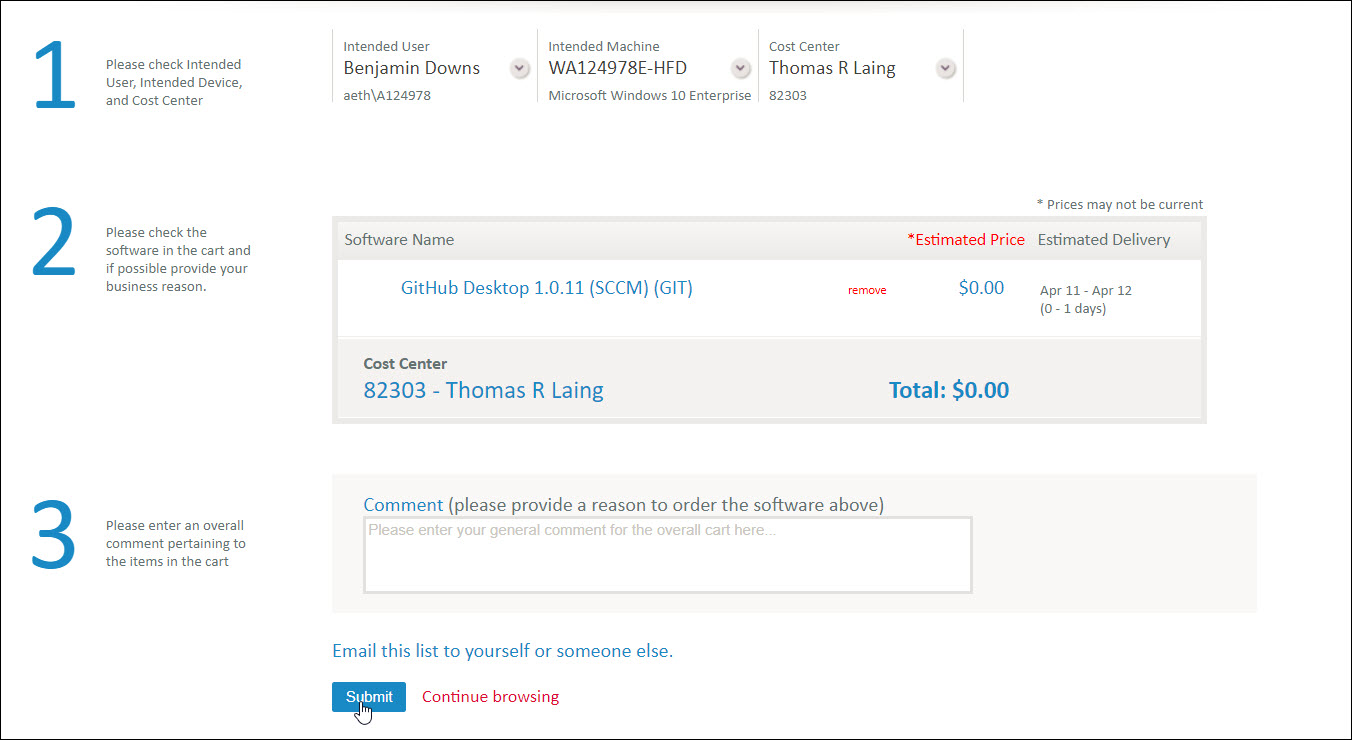
Add GitHub Desktop to cart



Checkout



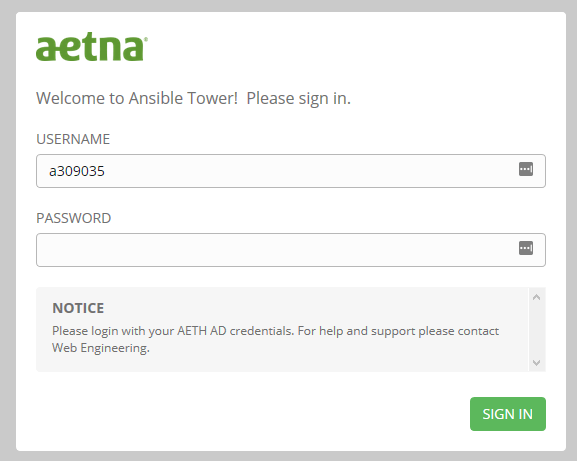
Confirm Purchase



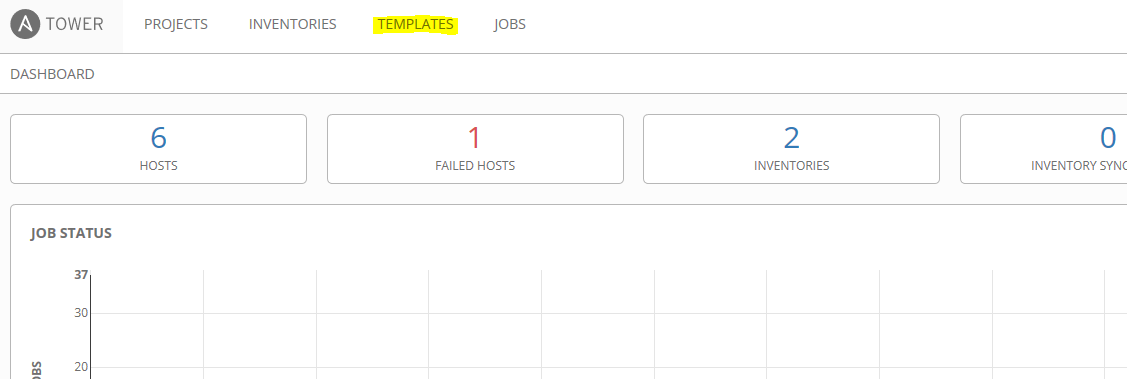
STEP 3: Run Pipeline Creation Scripts

* [AeTH](https://github.aetna.com/pages/IaC/Docs/gettingstarted/?tabs=tabid-10%2Ctabid-1%2Ctabid-3%2Ctabid-5%2Ctabid-51#tabpanel_CeZOj-G++Q-2_tabid-3)

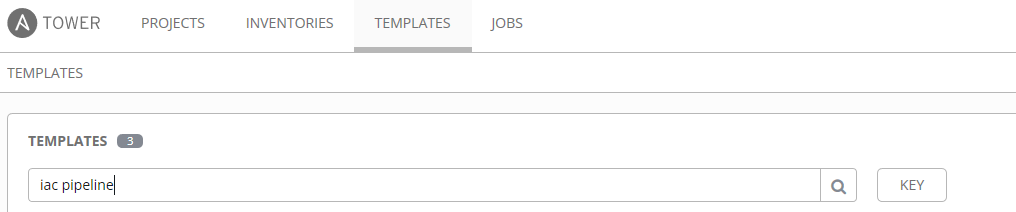
* [HeH](https://github.aetna.com/pages/IaC/Docs/gettingstarted/?tabs=tabid-10%2Ctabid-1%2Ctabid-3%2Ctabid-5%2Ctabid-51#tabpanel_CeZOj-G++Q-2_tabid-4)
* [AETH Pipeline Creation Jobs](https://tower.aetna.com/)
* Log in with your AETH credentials.



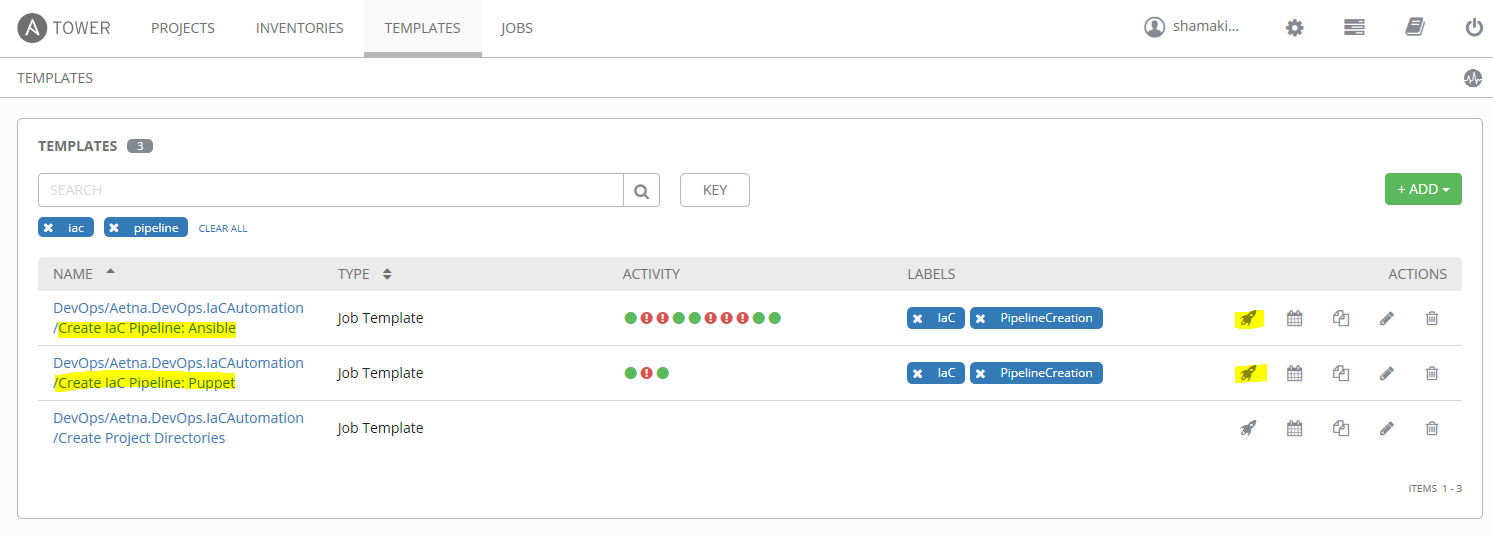
* Once logged in click "Templates" in the top navigation menu.



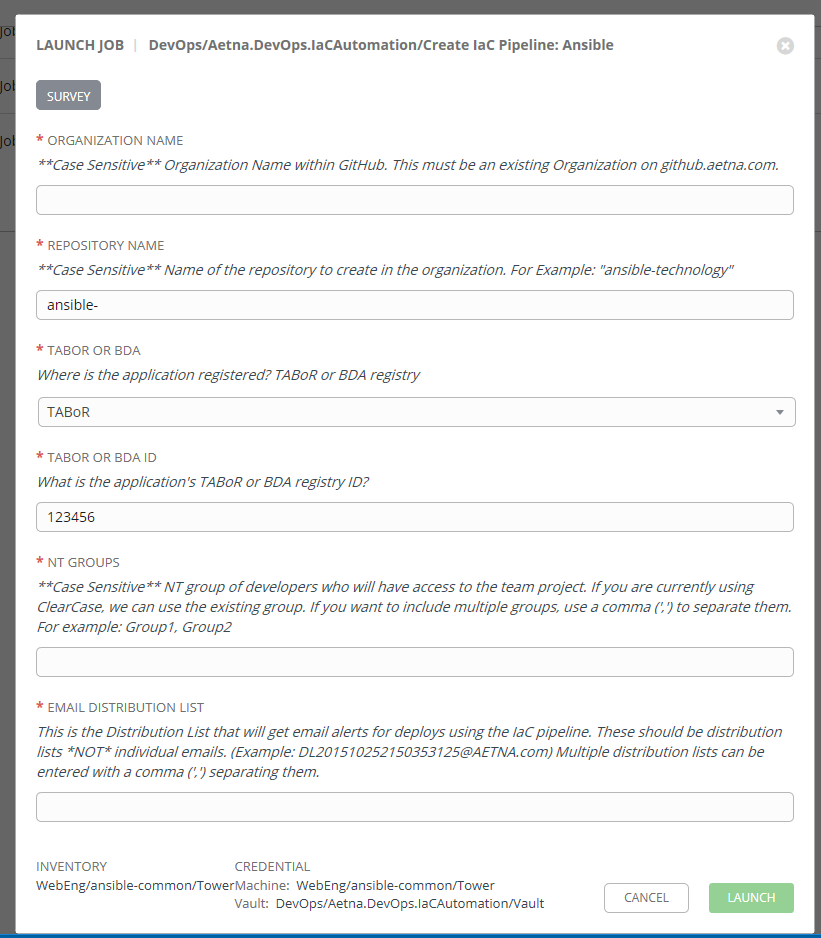
* On the Templates page search for "iac pipeline".



* This will limit the view to select Templates. Click the "Rocket Ship" icon next to the technology you wish to build a pipeline for.
* *Am I building/commissioning a new server or need to run a one shot?*
  + Yes - Click **/Create IaC Pipeline: Ansible**
  + No - Go to the next step below
* *Am I installing software and/or maintaning an installed product?*
  + Yes - Click **/Create IaC Pipeline: Puppet**
  + No - Ansible or Puppet may not be a solution, please contact the IaC team for further assistance



* Finally this will present you with a "Launch Job" dialog window. Answer the questions with the information you obtained from the Prerequisites Section above. Once done click "LAUNCH".



STEP 4: Git/GitHub Basics and Workflow

* [Command Line](https://github.aetna.com/pages/IaC/Docs/gettingstarted/?tabs=tabid-10%2Ctabid-1%2Ctabid-3%2Ctabid-5%2Ctabid-51#tabpanel_CeZOj-G++Q-3_tabid-5)

* [GitHub Desktop](https://github.aetna.com/pages/IaC/Docs/gettingstarted/?tabs=tabid-10%2Ctabid-1%2Ctabid-3%2Ctabid-5%2Ctabid-51#tabpanel_CeZOj-G++Q-3_tabid-6)

* [VS Code](https://github.aetna.com/pages/IaC/Docs/gettingstarted/?tabs=tabid-10%2Ctabid-1%2Ctabid-3%2Ctabid-5%2Ctabid-51#tabpanel_CeZOj-G++Q-3_tabid-7)

* [GitHub Website](https://github.aetna.com/pages/IaC/Docs/gettingstarted/?tabs=tabid-10%2Ctabid-1%2Ctabid-3%2Ctabid-5%2Ctabid-51#tabpanel_CeZOj-G++Q-3_tabid-8)

Adding/Pushing an Existing Project to Git (Command Line)

* It is recommended that you are acquainted with GitHub commands and syntax. Start by attending one of the free online training sessions available for [github](https://services.github.com/training/" \t "_blank). (You will need to create an account on github.com.
* Run the automation scripts
* Locally Navigate to the folder you want to upload to git
* git init initializes the folder as a git repository
* git add . stages all of the files in your folder
* git commit -m + "your commit message" to state why a certain update is being pushed
* git remote add + "your repository ulr"...copied down from github.aetna.com
* git push origin master will push all the code up to git. (you will need to enter your user id and password)
* git branch shows all of the local branches with an asterisk on the branch that you are on currently
* git branch -a shows all of the branches, local and remote
* git checkout -b creates a new branch based on the current branch that you are on

STEP 5: Code your Module

* [Ansible](https://github.aetna.com/pages/IaC/Docs/gettingstarted/?tabs=tabid-10%2Ctabid-1%2Ctabid-3%2Ctabid-5%2Ctabid-51#tabpanel_CeZOj-G++Q-4_tabid-51)

* [Puppet](https://github.aetna.com/pages/IaC/Docs/gettingstarted/?tabs=tabid-10%2Ctabid-1%2Ctabid-3%2Ctabid-5%2Ctabid-51#tabpanel_CeZOj-G++Q-4_tabid-52)

* [Python](https://github.aetna.com/pages/IaC/Docs/gettingstarted/?tabs=tabid-10%2Ctabid-1%2Ctabid-3%2Ctabid-5%2Ctabid-51#tabpanel_CeZOj-G++Q-4_tabid-53)

* [Bash](https://github.aetna.com/pages/IaC/Docs/gettingstarted/?tabs=tabid-10%2Ctabid-1%2Ctabid-3%2Ctabid-5%2Ctabid-51#tabpanel_CeZOj-G++Q-4_tabid-54)

* [PowerShell](https://github.aetna.com/pages/IaC/Docs/gettingstarted/?tabs=tabid-10%2Ctabid-1%2Ctabid-3%2Ctabid-5%2Ctabid-51#tabpanel_CeZOj-G++Q-4_tabid-55)

Folder Structure

Use the following folder structure - from here: [official Ansible documentation](http://docs.ansible.com/ansible/latest/user_guide/playbooks_best_practices.html#directory-layout)

roles/

common/ # this hierarchy represents a "role"

tasks/ #

main.yml # <-- tasks file can include smaller files if warranted

handlers/ #

main.yml # <-- handlers file

templates/ # <-- files for use with the template resource

ntp.conf.j2 # <------- templates end in .j2

files/ #

bar.txt # <-- files for use with the copy resource

foo.sh # <-- script files for use with the script resource

vars/ #

main.yml # <-- variables associated with this role

Naming Standards

* **Git Repos** - Follow a standard like "ansible-MeaningfulName" where the *name* is Pascal Cased and uses full words, when possible (as opposed to using acronyms).
* **Tower Assets** - Anything built in Tower (playbook, credentials, projects, etc) should be named like:  
        <github org>/<github repo>/<asset name>  
  For example, a playbook which resides in the IaC github organization and the ansible-test repository which installs the Apache web server might have a job template called: IaC/ansible-test/Install Apache.
* **Task Names** - Regular english sentences (e.g. "Update the configuration.")
* **Variables** - Generally follow Python-like naming. This means all lowercase and underscores to separate words. For example this\_is\_an\_example. For

# Production Deployment [Non-PCR Process]

The below document provides insight on how the development team will be using the new Non-PCR (aka RITM) process to schedule a Prod deploy in Octopus.

##### IMPORTANT

Make sure you create a **NEW** release & also make sure it is ready for Production Deploy in Octopus (Meaning it has been deployed to all lower environments). **Any release originally dated before 2/28/2019 is not compatible with this new process! Please create a new release in Octopus and promote it through your non-prod environments before attempting to use this new process**

**Overview of Steps to Setup:**

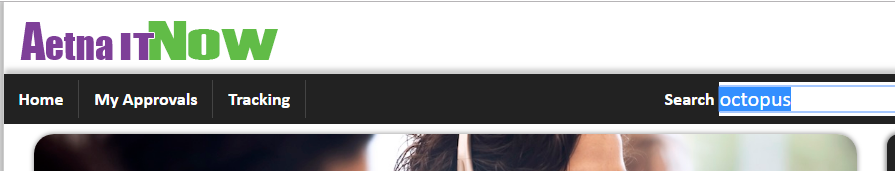
1. Have your Octopus release promoted through any required non-prod environments so its ready to be deployed to your production environment(s).
2. Create a ServiceNow change ticket for the Production Release.
3. Submit an AetnaITNow Prod Deploy Request (RITM Request). The development team does not need to wait for the change ticket to be approved to submit this request. Just make sure the ServiceNow change ticket is approved/scheduled **at least 10 minutes before scheduled deploy time**!

## Submit an AetnaITNow Prod Deploy Request (RITM Request)

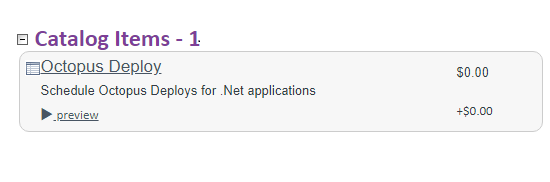
##### NOTE

RITMs can't be scheduled more than 30 days in advance. Also, make sure you give **at least 10 minutes** between getting the ServiceNow change ticket approved/scheduled and the RITM's "Requested Deploy Start Time". More on this below.

1. Navigate to [AetnaITNow](https://aetnaprod1.service-now.com/itnow" \t "_blank) and search for octopus in the search box.



1. Click **Octopus Deploy** to start the process of scheduling your Prod deployment in Octopus.



1. Select **Octopus Instance** and type - AETH. Other options include *HEH* for the HealthTheHost environment.



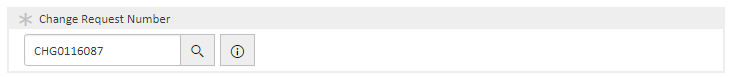
1. Type the exact Octopus **Project** Name.



1. Select the production **Environmnet** this deploy will be going into.



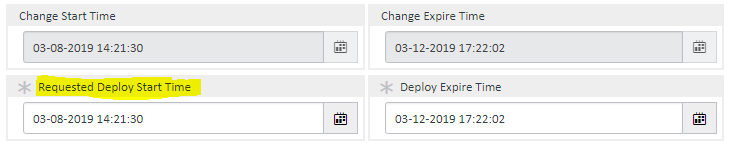
1. Type the **Change Request Number** you have submitted for this production deployment. NOTE: If you don't see any change requests in this field, make sure you have an [ITIL Role](https://aetnaprod1.service-now.com/itnow/catalog.do?v=1&uri=com.glideapp.servicecatalog_cat_item_view.do%3Fv%3D1%26sysparm_id%3De0db047fdba4fe809979f1e51d961957%26sysparm_link_parent%3D9874b72e6fefe10007a5f00dba3ee48e%26sysparm_catalog%3De0d08b13c3330100c8b837659bba8fb4&sysparm_document_key=sc_cat_item,e0db047fdba4fe809979f1e51d961957).



1. The above entered change request ticked will auto populate the **Change Start Time** and **Change Expire Time**. Those values are called the "Change Request Window". You do get an option to reschedule the deploy time, but only inside of the change request window. This can be done multiple times in case of deploy failures or Prod Issues.

##### NOTE

The grayed-out text boxes are the time entered in the ServiceNow change request ticket, which cannot be modified. The "**Requested Deploy Start Time**" **must** be used if the current time is later than the start time of the change ticket. If you want your migration to start at a specific time within the change ticket time you would specify it in the Requested Deploy Start Time field. If you find you need a fix or redo you must put in either the current time or a later time (must be within change request times). **The "Requested Deploy Start Time" MUST be the current time or later or it will fail.** Also, the ServiceNow change ticket must be approved/scheduled **at least 10 minutes before** the "**Requested Deploy Start Time**". If this isn’t followed, a new RITM will need to be submitted.



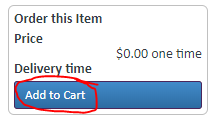
1. Select the **Release ID** that needs to be deployed to production.

##### NOTE

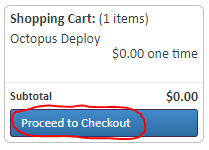
This Release ID **must** be ready to be deployed to the production environment! Meaning it has to have first successfully gone through the prerequisite non-prod environments first in Octopus Deploy. **Release ID's created before 2/28/2019 are not supported by this process! Please use the legacy PCR process for those.**



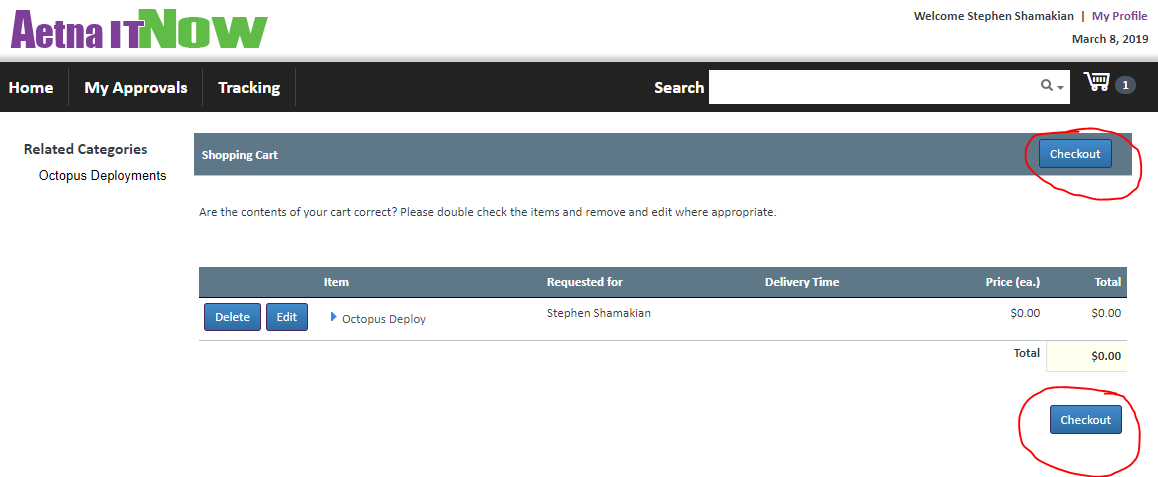
1. [Optional Step] If you have multiple deploys for same change request number, you can use **Add to Cart** to add them one after the other. Repeat the process by filling out the form and clicking **Add to Cart** for each Octopus Project/Release.



1. Once done, click on the **Proceed to Checkout** button to review your cart.



1. Click the **Checkout** button to submit the order and create the RITM(s).



## Deploying With Octopus

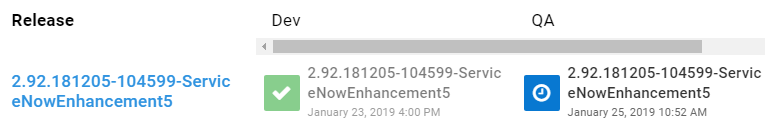
##### IMPORTANT

If the ServiceNow change ticket is already approved/scheduled at time of submitting the RITM request, it will schedule the deploy right away in Octopus (you will see this in your project's overview page in Octopus). Otherwise, it will wait until the ServiceNow change ticket is approved to schedule the deploy in Octopus. **Make sure the ServiceNow change ticket is approved atleast 10 minutes before the scheduled deploy time!**

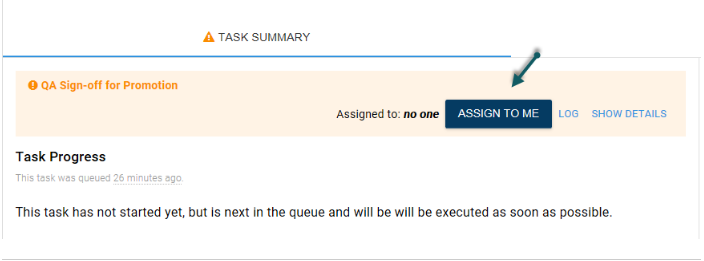
1. In [Octopus Deploy](xref:OctopusDeploy), check to see if the deployment is scheduled.

##### NOTE

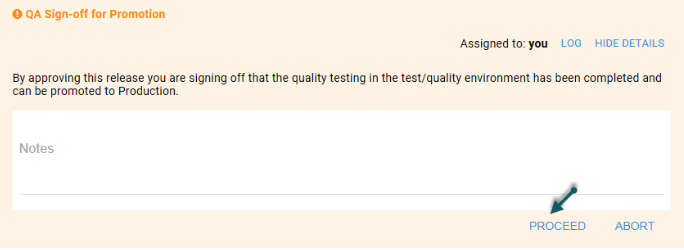
Image shows QA as scheduled, since this is just a demonstration this normally would be your Prod or Prod2 environment.



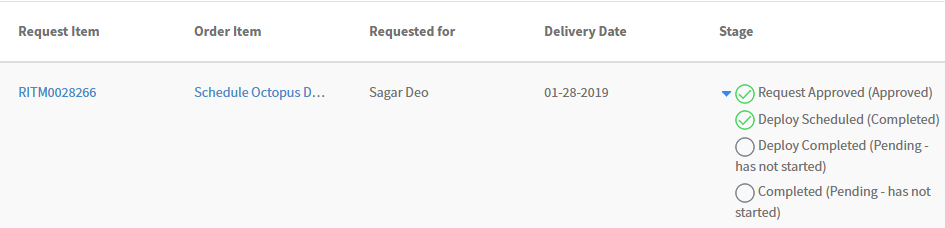
1. Once your deployment has started at the time mentioned in Octopus an email will be sent to the QA approvers group and a box will appear like this on the deployment page.



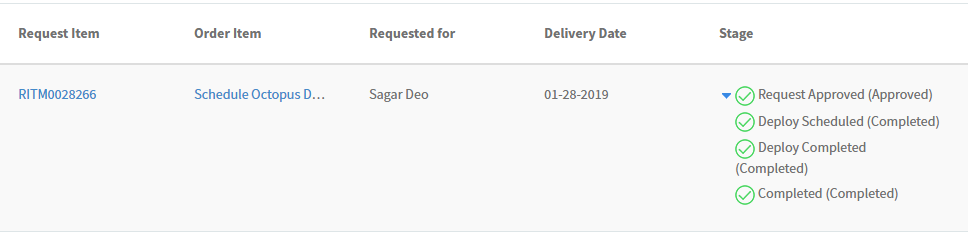
1. You will need to click **Assign to me** and optionally enter in any notes and finally click **Proceed**.



1. You can also track the status of Deploy in AetnaITNow.



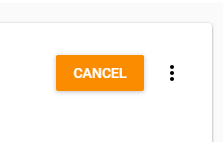
1. Once the Deploy is complete, the status will change to **Completed**.



## Cancelling A Prod Deploy

You can now self-cancel production deployments inside of Octopus Deploy. Look for the large orange "Cancel" button (top right of the screen when viewing the deployment). You can cancel at any time; while its scheduled/pending, waiting approval or durring the deploy.

* This is what the cancel button looks like:



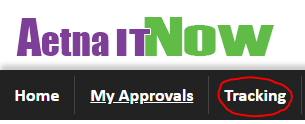
## Emergency Changes

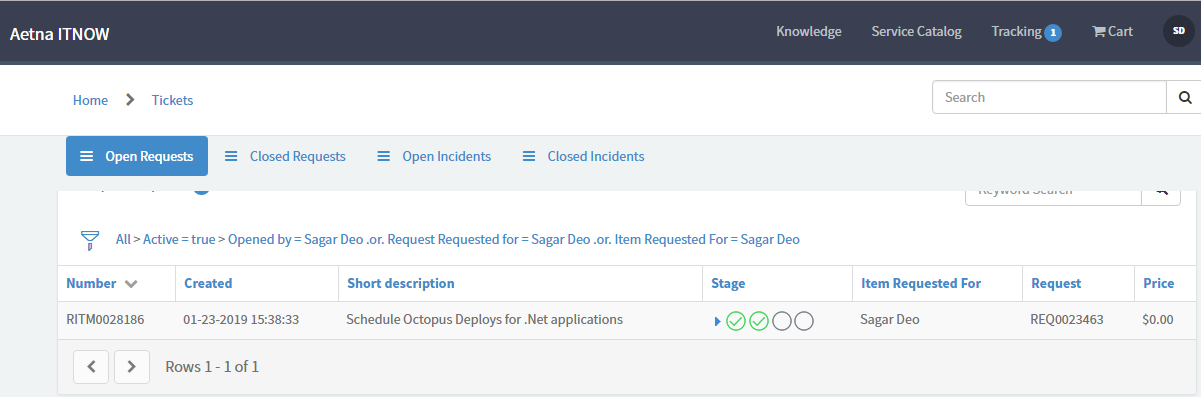
If you require an Octopus deploy for an emergency move you can create an Emergency Change ticket and create a RITM request for an automated deploy **without the Emergency ticket being scheduled/approved** to get the code in to production. You will then have 48 hours to make sure the emergency ticket gets approved and scheduled.

## Searching for an RITM request in AetnaITNow

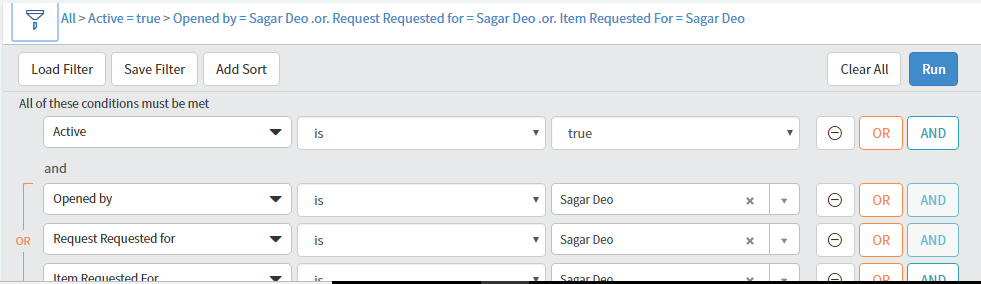
If you wish to search for the Octopus RITM request that you submitted, follow the below steps:

1. Navigate to [AetnaITNow](https://aetnaprod1.service-now.com/itnow" \t "_blank)
2. Click on **Tracking** to see all open/closed requests submitted by you





1. You can modify the **search filter** to look for specifc RITM



# About ansible tower.

# Getting Access

Access to Ansible Tower is controlled by Active Directory. Anyone with an active account in Active Directory is able to access Ansible Tower. The URLs for Tower are:

| **Environment** | **URL** |
| --- | --- |
| AETH | [https://tower.aetna.com](https://tower.aetna.com/) |
| HEH | [https://tower.healthehost.com](https://tower.healthehost.com/) |

Note: Please use firefox or Google Chrome for access to Tower. The Aetna build of Internet Explorer does not work well for rendering screens in Tower.

The login screen for Tower will look like this: 

On this screen enter your Active Directory username and password for the appropriate environment.

Once logged in you may or may not be able to access assets within Tower. This is due to a feature in Ansible Tower called [Roles Based Access Control](https://docs.ansible.com/ansible-tower/3.2.4/html/userguide/security.html#rbac-ug) (aka RBAC). Each asset (a credential, a job template, a workflow, etc) has roles associated with it which control who can see, use or edit the asset. Access to the various roles are control by Ansible Tower [teams](https://docs.ansible.com/ansible-tower/latest/html/userguide/teams.html). Teams are collections of Ansible Tower [users](https://docs.ansible.com/ansible-tower/latest/html/userguide/users.html).

The first time you authenticate to Tower with your Active Directory credentials a user will automatically be created for you in Ansible Tower. Your user ID will be the same as your Active Directory ID and your email address will also be imported and associated with your user. In addition, if you are a member of a group in Active Directory and that group has team membership in Tower, your Tower user will automatically be associated to the team in Tower.

Each time you authenticate to Tower your team association will be adjusted to match your Active Directory groups so your access to assets in Tower is dynamic.

In addition, there is a team called everyone which anyone who logs into Tower will be granted access to. There are a limited number of items which are exposed to everyone.

../../../images/ansible/logout.png To log out of Ansible Tower, click on the logout (power) button in the upper left corner of the Ansible Tower screen.

# Setting up a Tower Project

The setup of a [project](https://docs.ansible.com/ansible-tower/latest/html/userguide/projects.html) in Tower is an automated process and is completely controlled by the [Pipeline build process](https://github.aetna.com/pages/IaC/Docs/iacservices/gettingstarted/index.html#step-3-run-pipeline-creation-scripts).

During the pipeline creation process a Tower project will be setup with a name matching your GitHub project in the format <GitHub Organization>/<GitHub Project>. For example, if your Organization was WebEngineering and your project was ansible-websphere, the pipeline would create a Tower project called WebEngineering/ansible-websphere.

Note: This project will only be setup in the lowest environment of Ansible Tower. As you migrate your code through the pipeline that migration will add the project to further Tower environments.

The pipeline will also make sure that the Active Directory groups you specified for access to the GitHub project are also created in Ansible Tower as [teams](https://docs.ansible.com/ansible-tower/latest/html/userguide/teams.html) and that those teams have read access to your project.

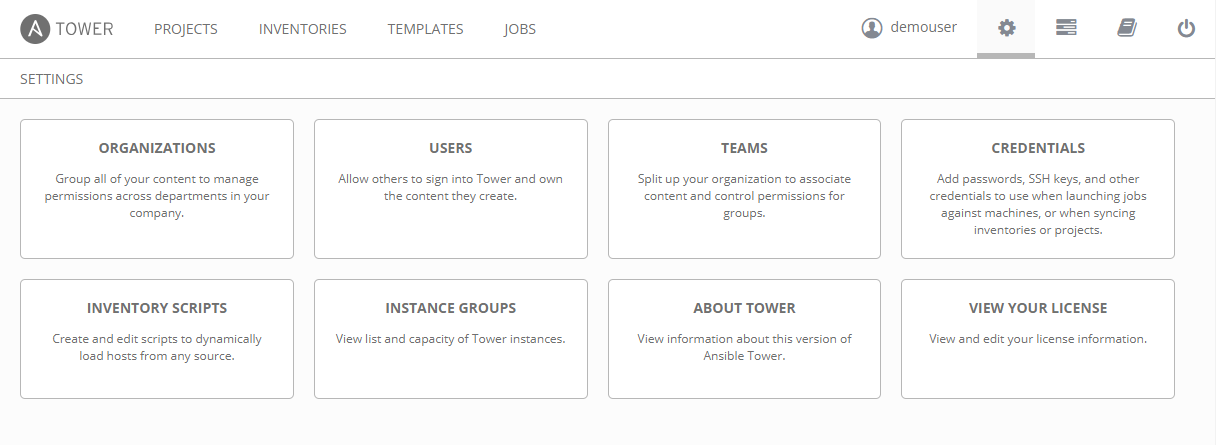
Once your project is setup by the pipeline process it does not need to be altered in any way. No teams will need permissions to your project other than the teams that will be creating job templates from the files in your project. This is why you only have the view [role](https://docs.ansible.com/ansible-tower/latest/html/userguide/security.html#rbac-ug) on your project. If, for any reason, your project needs to be altered please contact [Ansible Support](mailto://AnsibleSupport@AETNA.com)

# Setting up Credentials

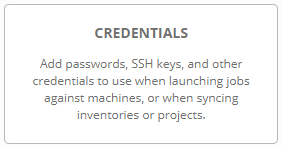
[Credentials](https://docs.ansible.com/ansible-tower/latest/html/userguide/credentials.html) allow Ansible Tower to access resources such as: machines, network devices, vault files, cloud providers, etc. Credentials are stored securely in Ansible Tower an no one will be able to access the secrets (passwords, ssh keys, etc) once they are entered into Tower (not even the owner of a credential).

# Add a credential

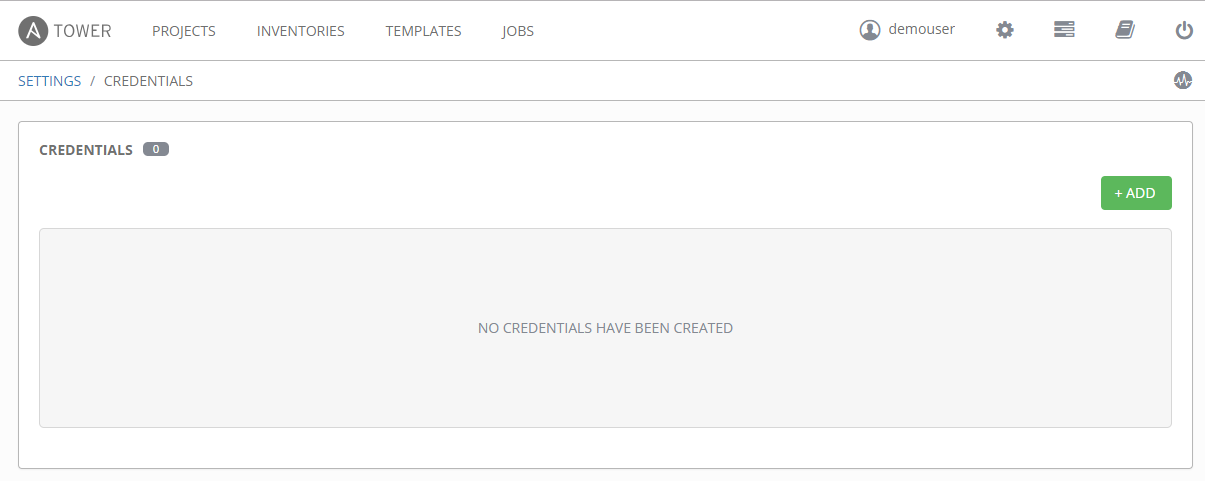
../../../images/ansible/settings.png To build a credential click on the Settings (gear) icon located in the upper right screen of Tower. This will take you to the settings screen:



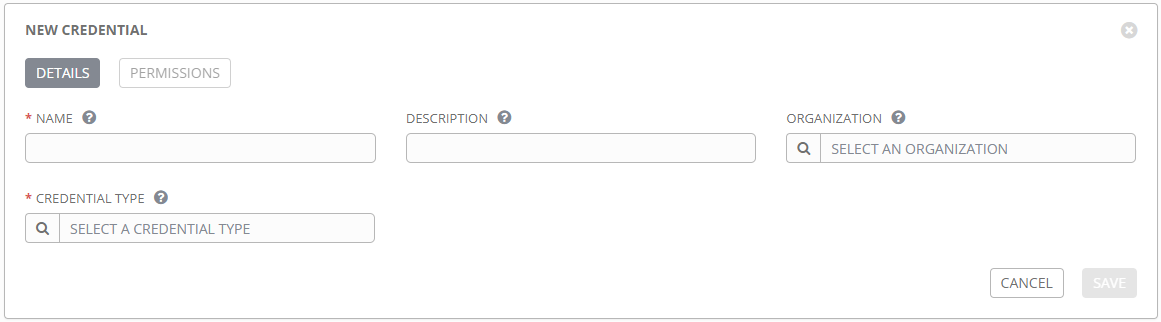
On this screen click on the "Credentials" box:



On the credentials screen you may or may not see some existing credentials:



../../../images/ansible/add.pngClick on the green Add button on the right hand side. This will open a new section above the list of credentials for adding a new credential:



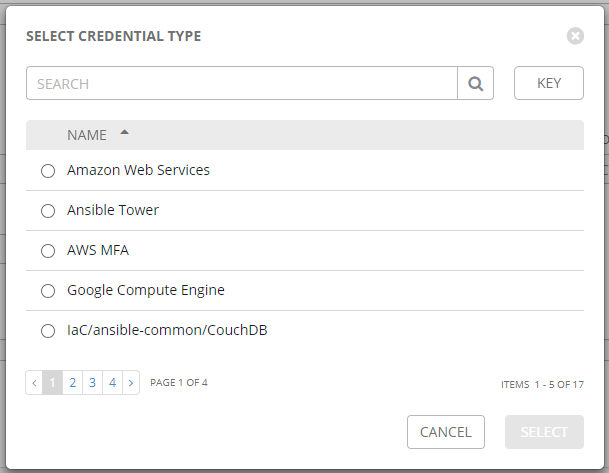
On this screen:

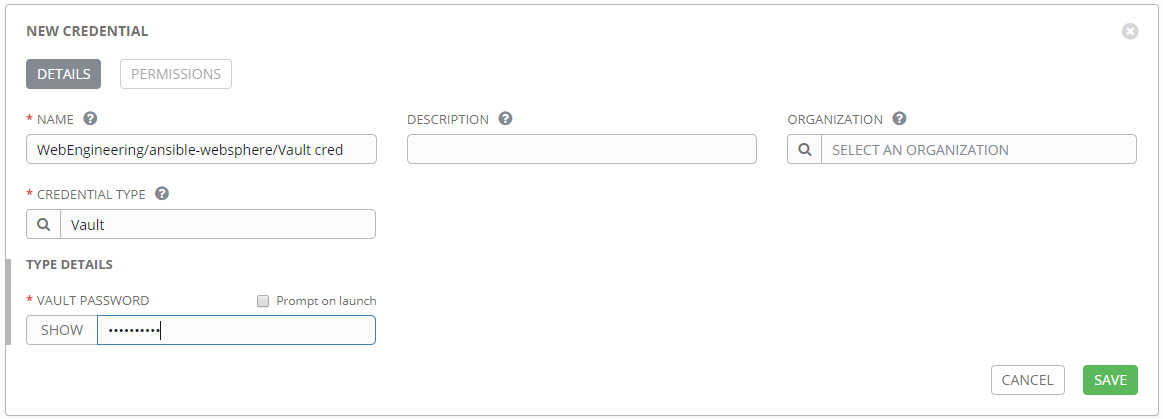
Note: Fields with a red asterisk (\*) are required fields

* Enter the name of the credential. This must be in the format <project name>/<credential name>. For example, if your project is WebEngineering/ansible-websphere and your credential will be called "Vault cred" the name of your credential will be: WebEngineering/ansible-websphere/Vault cred.
* Enter an optional description.

Note: By default, you will not be able to add an Organization to your credential. The credential will initially be for personal use. See the [raven](https://github.aetna.com/pages/IaC/Docs/iacservices/ansible/tower/raven.html#permissions) page for information on how to add permissions to the credential.

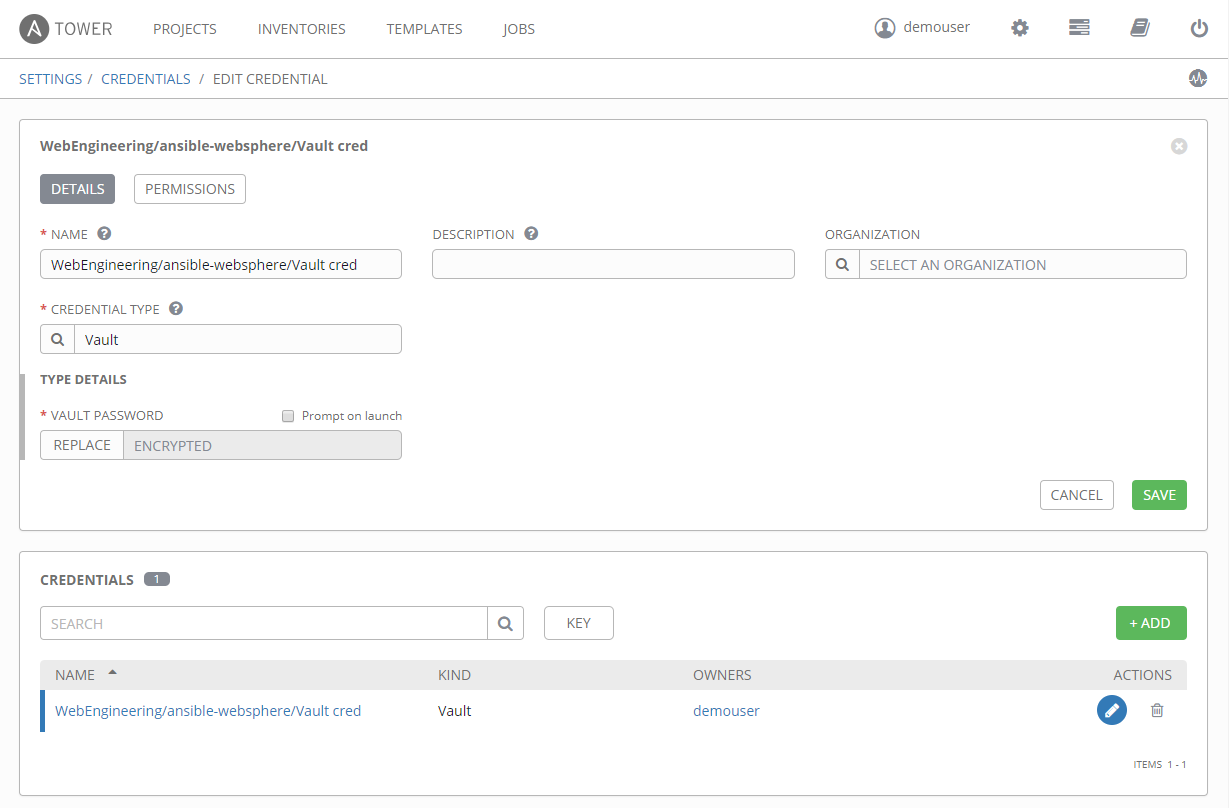
../../../images/ansible/spyglass.png Next select the credential type by clicking on the spyglass icon. This will open the credential type selection screen:

 On this screen you can use the search bar to search for a credential type or use the page selection to scroll through the different types of credentials. Continuing with our example, we will be selecting a credential type of Vault. Once selected, this will open a new section on the screen to ask for additional details based on the credential type. For a vault credential, the only additional information is "Vault Password":



Note: the password I entered is added with asterisks

../../../images/ansible/save.pngOnce all of your credential and credential type information has been properly entered click the green save button. This will bring you to the edit screen of your credential and will add the credential to the list of credentials listed below the edit screen:



Note: The Vault Password field has changed from asterisks to the word "ENCRYPTED". This indicates that the field has been securely stored in the database and can not be retrieved by anyone ever again.