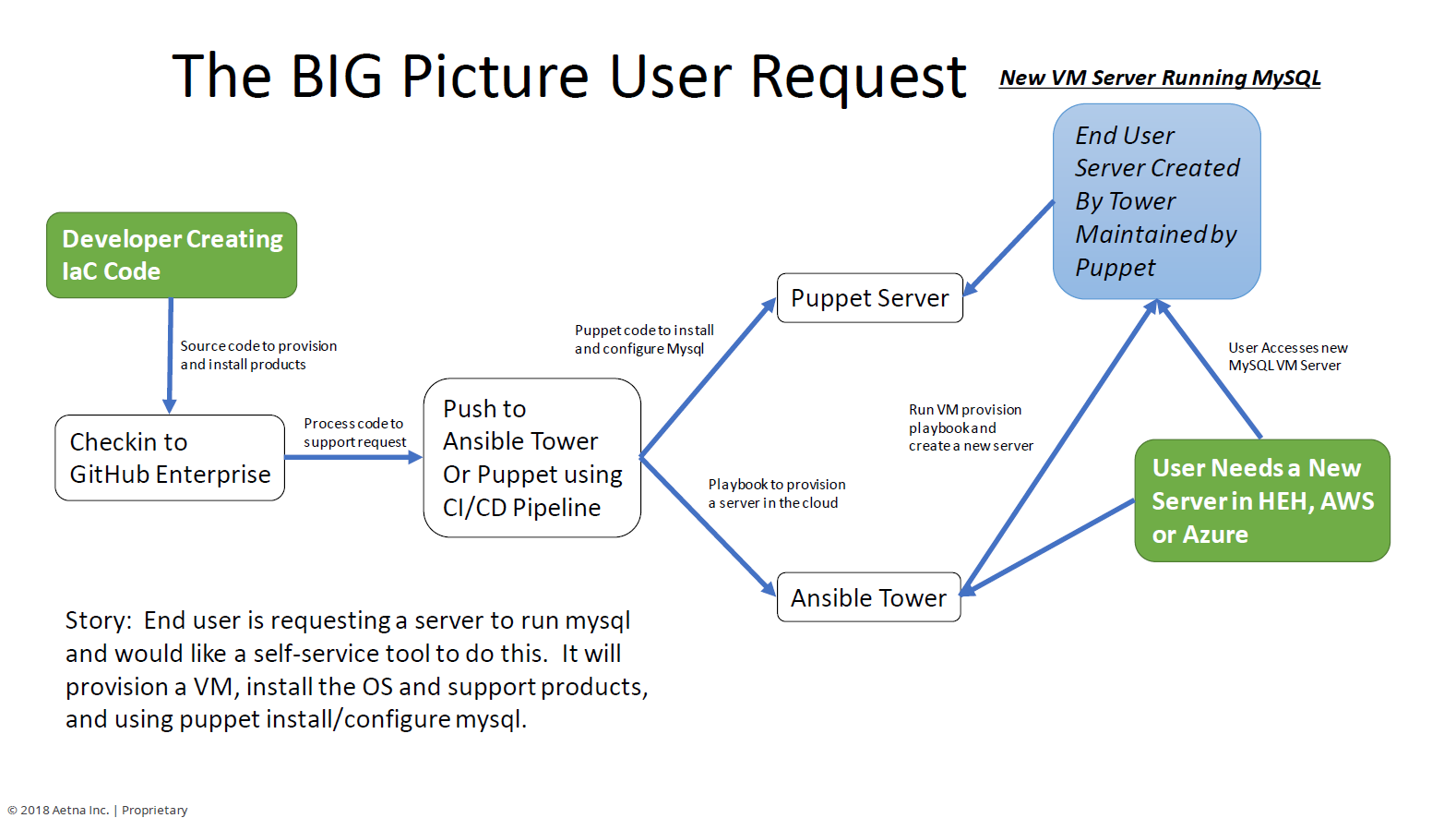
Getting Started

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).

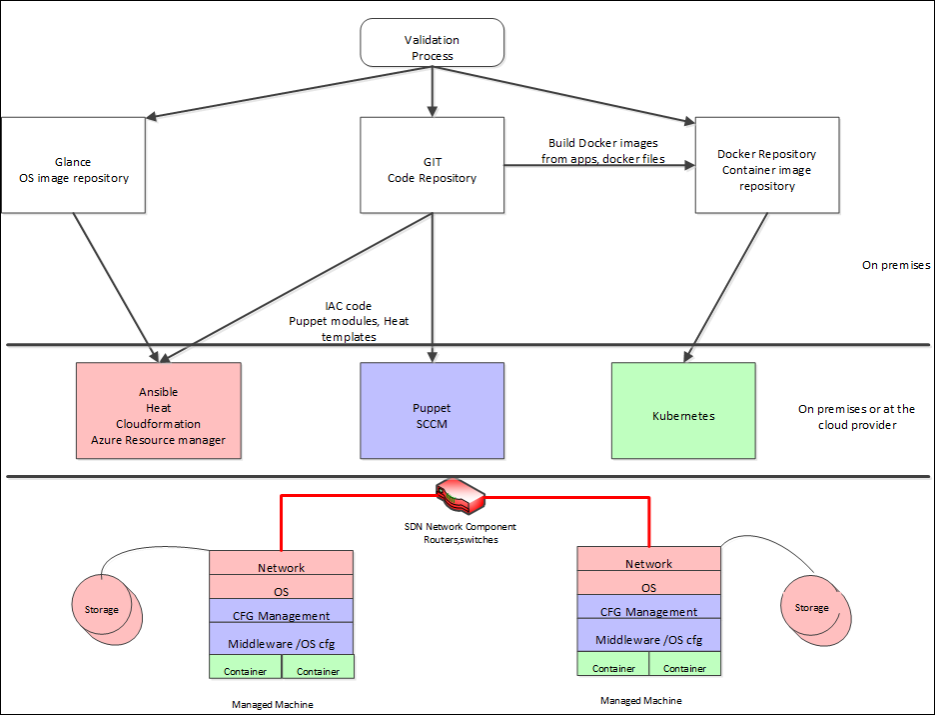
AeTH and HeH Pipelines

* 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

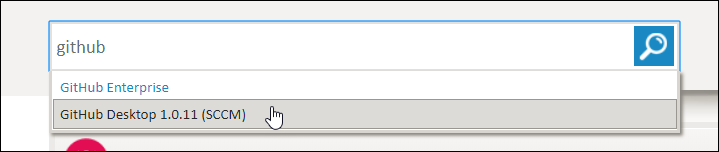
STEP 1: Get license for GHE

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

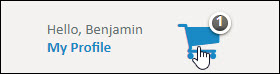
* Visit the Aetna [Appstore](https://appstore.aetna.com/) and, as of this writing April 2018, search for "GitHub Desktop 1.0.11 (SCCM)".



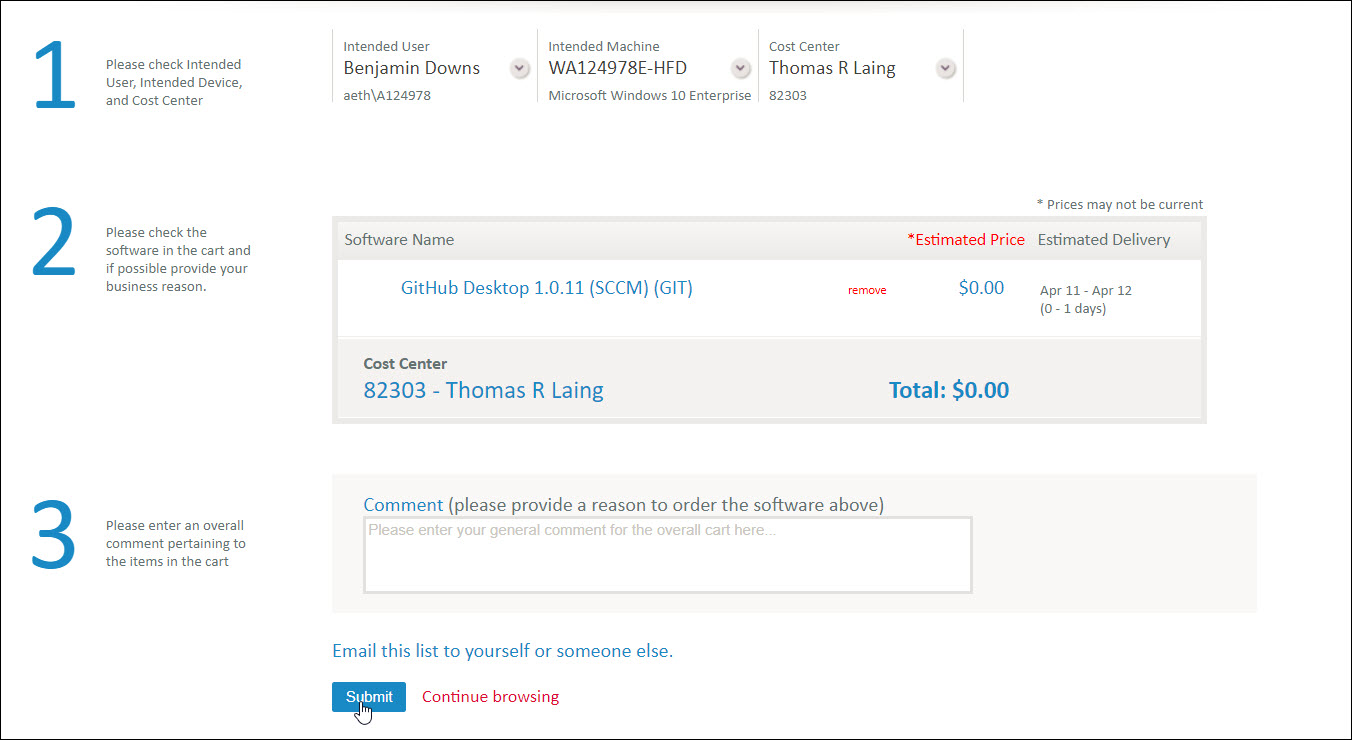
Add GitHub Desktop to cart



Checkout



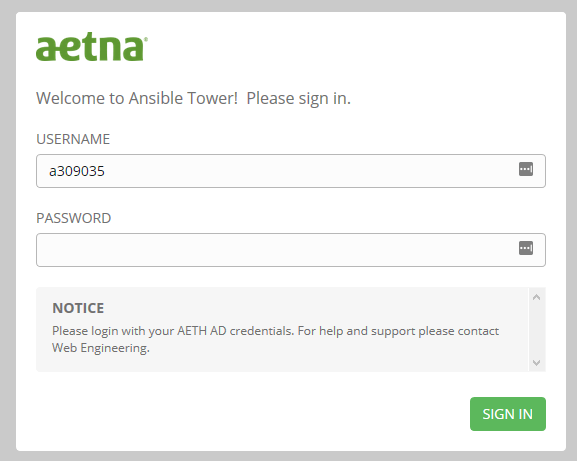
Confirm Purchase



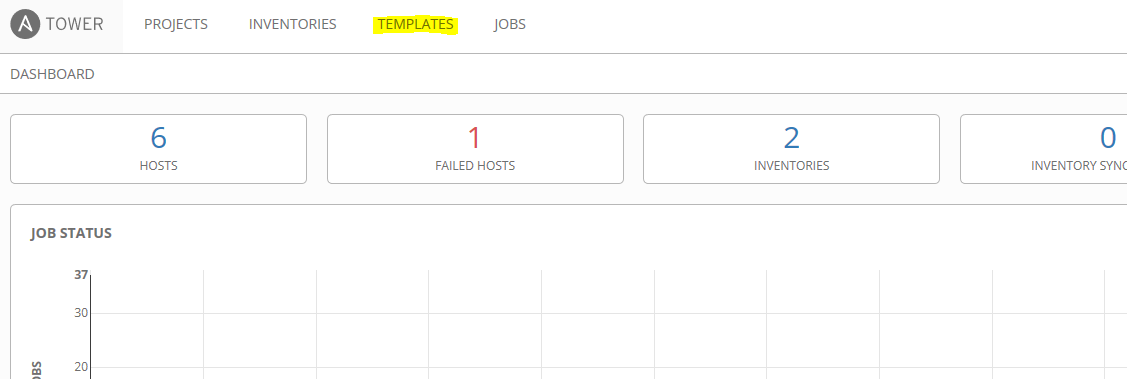
STEP 3: Run Pipeline Creation Scripts

* [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)

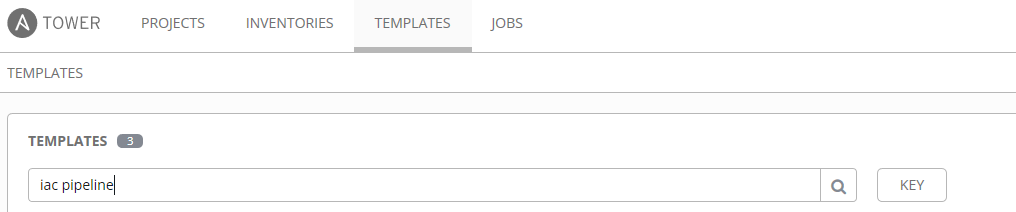
* [HeH](https://github.aetna.com/pages/IaC/Docs/gettingstarted/?tabs=tabid-10%2Ctabid-1%2Ctabid-3%2Ctabid-5%2Ctabid-51#tabpanel_CeZOj-G++Q-2_tabid-4)
* [AETH Pipeline Creation Jobs](https://tower.aetna.com/)
* Log in with your AETH credentials.



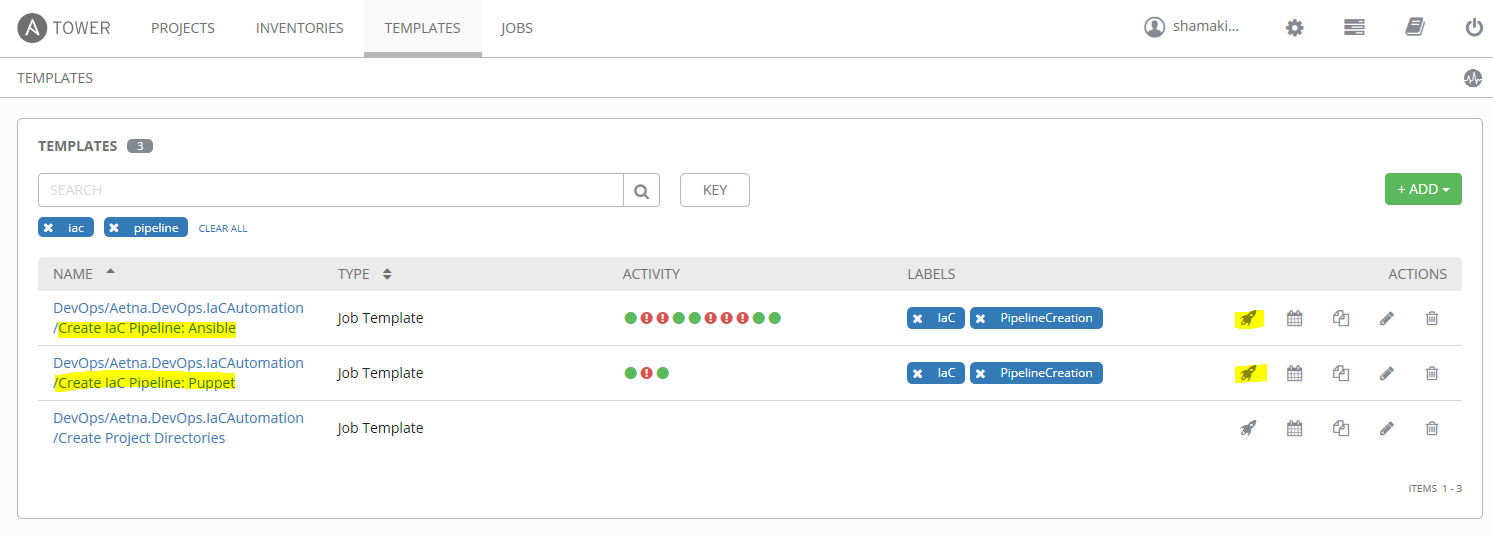
* Once logged in click "Templates" in the top navigation menu.



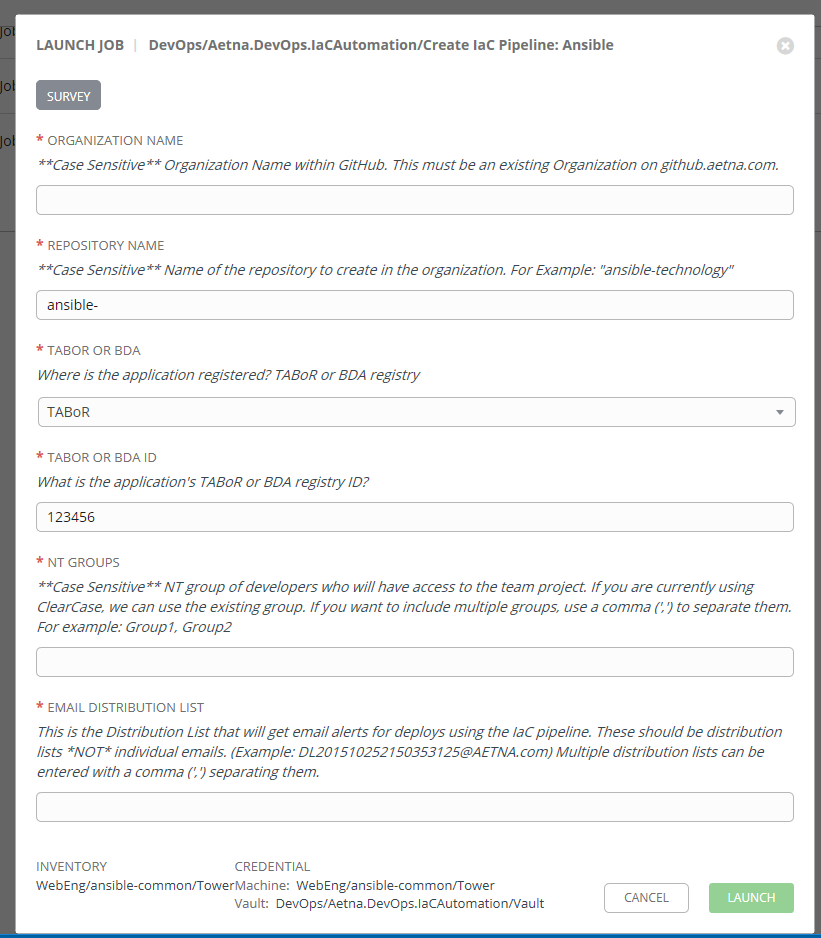
* On the Templates page search for "iac pipeline".



* 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/). (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

Tabs

All YAML files uses 2 spaces for a tab instead of 4 spaces or a TAB

--

***More information:***[***Ansible Best Practices***](https://github.aetna.com/pages/IaC/Docs/moduledevelopment/BestPractices/ansible.html)