Ansible Tower

[Video description begins] Topic title:Ansible Tower. Your host for this session is Sven Batalla. [Video description ends]

Ansible Tower, which was formerly referred to as AWX, can be simplistically considered as a wrapper around the Ansible Engine. Ansible Tower is a web-based solution whose one main job is to ease Ansible usage for users. Now Ansible Tower makes Ansible much easier to use for DevOps teams or really anyone that needs to interact with Ansible. It simplifies the usage and has been designed to be the main hub for all of your Ansible automation needs. In the early days of Ansible, the main piece of feedback that users gave was that there was no easy way to interact with it. Everything you did with Ansible was through command line interfaces. Well, with Ansible Tower there's now much easier user interface that lets anyone interact with Ansible with a much smaller barrier to entry. Now, one of the main features of Ansible Tower is that it allows you to finely control access to who can control and access what.

This even includes things like allowing the sharing of SSH credentials without someone actually being able to see or transfer those credentials. And since Ansible Tower includes that fancy web-based user interface, you're able to manage things graphically, rather than through text-based command line requests. In fact, a common activity is to manage your inventory through Ansible Tower or even synchronize your inventory with a wide range of cloud providers, like AWS, Azure, and so on. Ansible Tower also logs all of your jobs, integrates with LDAP and has loads of other features. And while you can do that through the web-based user interface, it doesn't have to be done that way. Even have the option of using a browsable REST API or yes, the command line.

The idea is that sure, users can browse Ansible Tower, but the REST API or command line means that applications like say Jenkins can use Ansible Tower as well. The Ansible Tower is to dominate interface for interacting with Ansible in a user-friendly way. So, let's go over some of the features of Ansible Tower. First is the automation dashboard. The automation dashboard provides a heads-up NOC-style dashboard that gives you all of the information that you need about the current state of your Ansible environment. This also includes everything that's currently going on, so that can be the network state, recently run jobs, and so much more. The point of the dashboard is for you to be able to see what's going on at a glance and at any time. Another feature of Ansible Tower is role-based access control or RBAC.

With RBAC you can create the needed separation between users and resources, so users can quickly, easily, and safely browse Ansible Tower and only work on systems that they have access to. It also means that if you've configured roles correctly, no unauthorized user can do anything that would impact your system in a negative way. Ansible Tower also includes delegation and self-service features. What that means is that you can use Ansible Tower to easily import users, entire teams, and even the whole organization from your SAML, Active Directory, or LDAP directories, and then delegate access from there. This can include delegating automation through simple question and answer configuration, automatic scheduling, and more. In other words, rather than setting up your own users in Ansible, you can delegate to your corporate user repository and even manage access from there.

Adding and removing users can then be managed through your corporate directory. Another feature of Ansible Tower is automation workflows. With Ansible Tower workflows, you can chain together all kinds of different tasks to create logical workflows. You can model your processes and deployments right within Ansible Tower. A common example is creating a workflow for how to recover from failures, and you can even chain workflows to each other. Now, earlier I mentioned that Ansible Tower includes alternatives to the web-based user interface and the most common alternative is the REST API. Anything you can do in the Ansible Tower user interface you can do in the REST API. Where the API is handy is if you want to integrate with other systems into Ansible Tower.

So, for example, if you wanted to create a system that could remain informed about the status of your various automation workflows, you could connect to the API to gather that information on a continuous basis. Common implementations of the REST API are for things like integration with continuous delivery systems or even for the purposes of ticketing. Now another feature of Ansible Tower is its auditing and logging capabilities. All the automation done within Ansible is securely logged and stored to provide you with a full audit trail of what happened with each run. You even have the ability to flow logs to third-party systems. And that's generally something you do if you want to gather analytics and perform a detailed analysis and correlation activity for the purposes of things like alerting and other forms of managing.

And remember that Ansible Tower is effectively a wrapper around Ansible. An Ansible helps you manage inventory. So, that means that Ansible Tower is able to help you manage your inventory in a much more user-friendly and visual way. It even helps you to sync your inventory with cloud providers, add and remove inventory, and even help you gather insights. And one last feature of Ansible Tower is the ability to scale-out clustering. You can actually connect multiple Ansible tower nodes into a single tower cluster so that you can add redundancy, capacity, and high availability. This includes the ability to reserve capacity. You can add push button deployment and scaling when you're running the Red Hat OpenShift container platform.