Monitoring and Visualizing Ansible Tower Metrics using Prometheus and Grafana.

As you may be aware, Ansible Tower v >3.5 [1] includes several metrics that are accessible via the API v2. These metrics can be consumed by monitoring tools such as Prometheus and visualized through dashboards such as Grafana :)

Using these metrics, we can get insights into Ansible Tower’s telemetry. A few examples are as follows:

- Host count and license usage
- Memory and CPU consumption of each Tower instance
- The number of jobs running and their status
- Successfully completed jobs

Let’s do this :)

To access these metrics, you can make the following call to Ansible Tower API:

```
[root@prometheus ~]# curl -sk "https://towercloud.skynet.com/api/v2/metrics/" -H
"Authorization: Bearer [redacted]"
```

The output of the above call provides several interesting metrics together with a short description for each.

```
# HELP awx_system_info AWX System Information
# TYPE awx_system_info gauge
awx_system_info{ansible_version="2.8.4",external_logger_enabled="False",external_logger_type
="None",insights_analytics="False",install_uuid="9360e3ed-9e9b-4edb-8b42-b82971fe5aff",licen
se_expiry="81182832",license_type="enterprise",pendo_tracking="detailed",tower_url_base="htt
ps://towercloud.skynet.com",tower_version="3.5.2"} 1.0
# HELP awx_organizations_total Number of organizations
# TYPE awx_organizations_total gauge
awx_organizations_total 4.0
# HELP awx_users_total Number of users
# TYPE awx_users_total gauge
awx_users_total 6.0
# HELP awx_teams_total Number of teams
# TYPE awx_teams_total gauge
awx_teams_total 0.0
# HELP awx_inventories_total Number of inventories
# TYPE awx_inventories_total gauge
awx_inventories_total 2.0
# HELP awx_projects_total Number of projects
# TYPE awx_projects_total gauge
awx_projects_total 2.0
# HELP awx_job_templates_total Number of job templates
# TYPE awx_job_templates_total gauge
awx_job_templates_total 15.0
# HELP awx_workflow_job_templates_total Number of workflow job templates
# TYPE awx_workflow_job_templates_total gauge
awx_workflow_job_templates_total 6.0
# HELP awx_hosts_total Number of hosts
# TYPE awx_hosts_total gauge
awx_hosts_total{type="all"} 2.0
awx_hosts_total{type="active"} 2.0
# HELP awx_schedules_total Number of schedules
# TYPE awx_schedules_total gauge
awx_schedules_total 2.0
# HELP awx_inventory_scripts_total Number of invetory scripts
# TYPE awx_inventory_scripts_total gauge
awx_inventory_scripts_total 0.0
# HELP awx_sessions_total Number of sessions
# TYPE awx_sessions_total gauge
awx_sessions_total{type="all"} 0.0
awx_sessions_total{type="user"} 0.0
awx_sessions_total{type="anonymous"} 0.0
# HELP awx_custom_virtualenvs_total Number of virtualenvs
# TYPE awx_custom_virtualenvs_total gauge
awx_custom_virtualenvs_total 0.0
# HELP awx_running_jobs_total Number of running jobs on the Tower system
# TYPE awx_running_jobs_total gauge
awx_running_jobs_total 0.0
# HELP awx_instance_capacity Capacity of each node in a Tower system
# TYPE awx_instance_capacity gauge
awx_instance_capacity{instance_uuid="402f3c6e-1162-466d-b168-2cd3aebdd380"} 59.0
# HELP awx_instance_cpu CPU cores on each node in a Tower system
# TYPE awx_instance_cpu gauge
awx_instance_cpu{instance_uuid="402f3c6e-1162-466d-b168-2cd3aebdd380"} 2.0
# HELP awx_instance_memory RAM (Kb) on each node in a Tower system
# TYPE awx_instance_memory gauge
awx_instance_memory{instance_uuid="402f3c6e-1162-466d-b168-2cd3aebdd380"} 8.370778112e+09
# HELP awx_instance_info Info about each node in a Tower system
# TYPE awx_instance_info gauge
awx_instance_info{enabled="True",instance_uuid="402f3c6e-1162-466d-b168-2cd3aebdd380",last_i
solated_check="None",managed_by_policy="True",version="3.5.2"} 1.0
# HELP awx_instance_launch_type_total Type of Job launched
# TYPE awx_instance_launch_type_total gauge
awx_instance_launch_type_total{launch_type="manual",node="localhost"} 163.0
awx_instance_launch_type_total{launch_type="workflow",node="localhost"} 760.0
awx_instance_launch_type_total{launch_type="dependency",node="localhost"} 27.0
awx_instance_launch_type_total{launch_type="relaunch",node="localhost"} 8.0
awx_instance_launch_type_total{launch_type="scheduled",node="localhost"} 7.0
# HELP awx_instance_status_total Status of Job launched
# TYPE awx_instance_status_total gauge
awx_instance_status_total{node="localhost",status="successful"} 929.0
awx_instance_status_total{node="localhost",status="failed"} 33.0
awx_instance_status_total{node="localhost",status="canceled"} 3.0
# HELP awx_license_instance_total Total number of managed hosts provided by your license
# TYPE awx_license_instance_total gauge
awx_license_instance_total 500.0
# HELP awx_license_instance_free Number of remaining managed hosts provided by your license
# TYPE awx_license_instance_free gauge
awx_license_instance_free 498.0
```

This is all great, but what is required in order to be able to consume these metrics? The answer lies within authentication to Ansible Tower.
There are two methods for authentication to Ansible Tower: [2]

- Basic authentication (User and Password)
- OAuth2 Token Authentication.

In order to use any of these methods, we will need to have a Tower user.

In Ansible Tower 3.5, the following user profiles are allowed to access the metrics: [3]

- System Administrator
- System Auditor

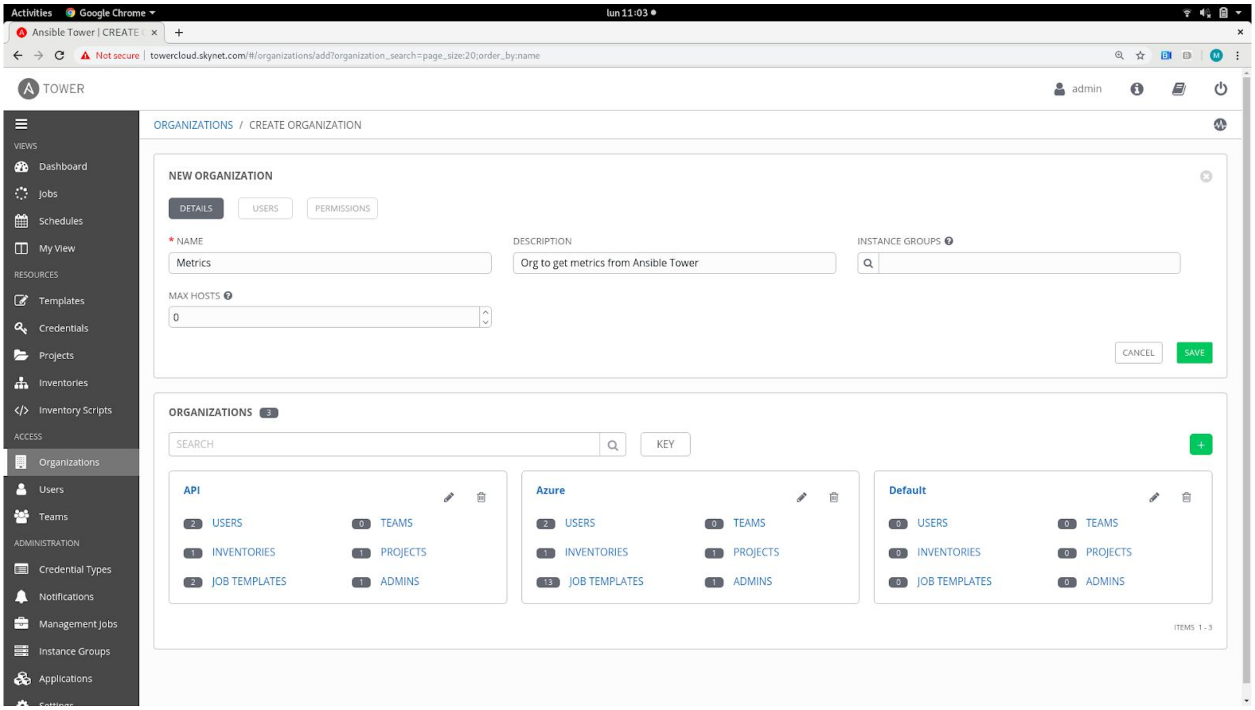
Normal users *are not allowed to* access the metrics through an API call.

In order to ensure that Ansible Tower metrics are accessed securely, we can dedicate a special user for this purpose. On Isolating this dedicated use, we need to consider the following:

- ONLY one organization should be used for this purpose. (Remember, all users must have an organization, except the System Administrator used to install the product.)
- We should only have one user with the System Auditor profile inside this organization.
- Only one personal OAuth2 token should be issued for the dedicated user.

So, let’s do this! :)

The first step is to create the organization, in this example we call it “metrics”. We’ll need to perform this as the System Administrator user.



ORGANIZATIONS / CREATE ORGANIZATION

NEW ORGANIZATION

DETAILS | USERS | PERMISSIONS

* NAME	DESCRIPTION	INSTANCE GROUPS
Metrics	Org to get metrics from Ansible Tower	
MAX HOSTS		
0		

CANCEL | SAVE

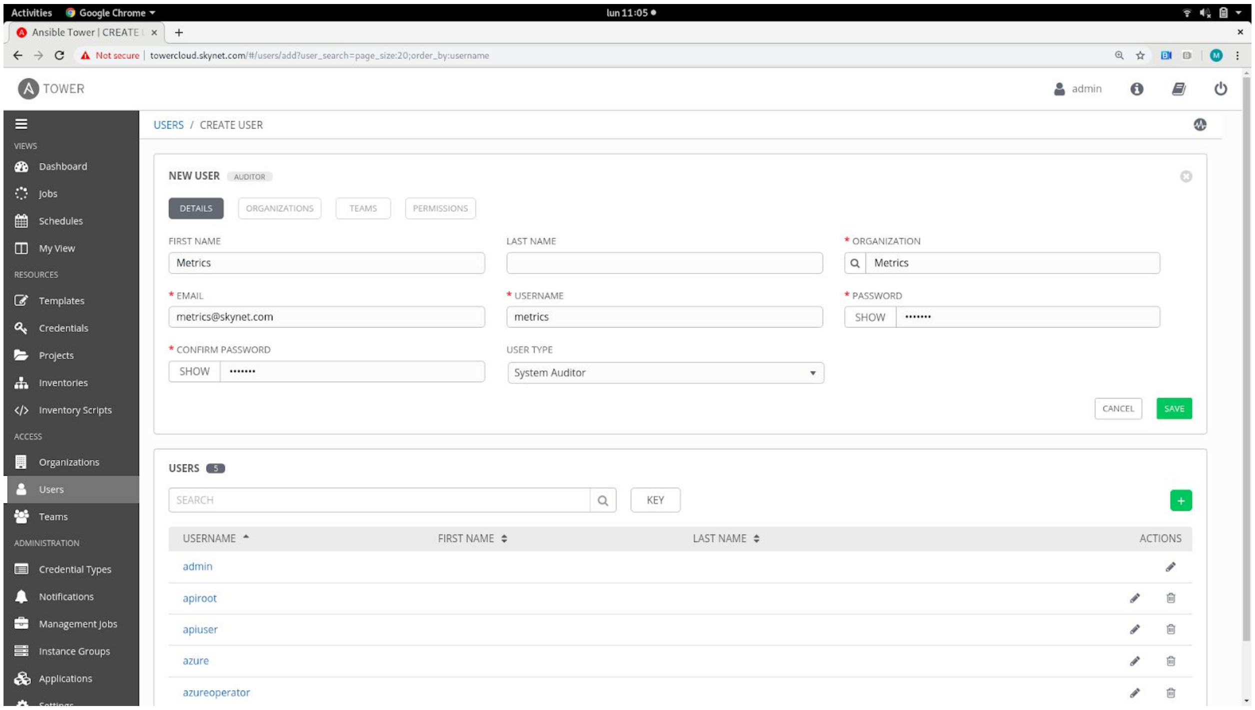
ORGANIZATIONS 3

SEARCH | KEY | +

	API	Azure	Default
USERS	2	2	0
TEAMS	0	0	0
INVENTORIES	1	1	0
PROJECTS	1	1	0
JOB TEMPLATES	2	13	0
ADMINS	1	1	0

ITEMS 1 - 3

Then we need to create a user called “metrics” with a System Auditor profile in the “metrics” organisation. We will do this in the same session login in as the System Administrator.



USERS / CREATE USER

NEW USER AUDITOR

DETAILS | ORGANIZATIONS | TEAMS | PERMISSIONS

FIRST NAME	LAST NAME	* ORGANIZATION
Metrics		Metrics
* EMAIL	* USERNAME	* PASSWORD
metrics@skynet.com	metrics	SHOW •••••••
* CONFIRM PASSWORD	USER TYPE	
SHOW •••••••	System Auditor	

CANCEL | SAVE

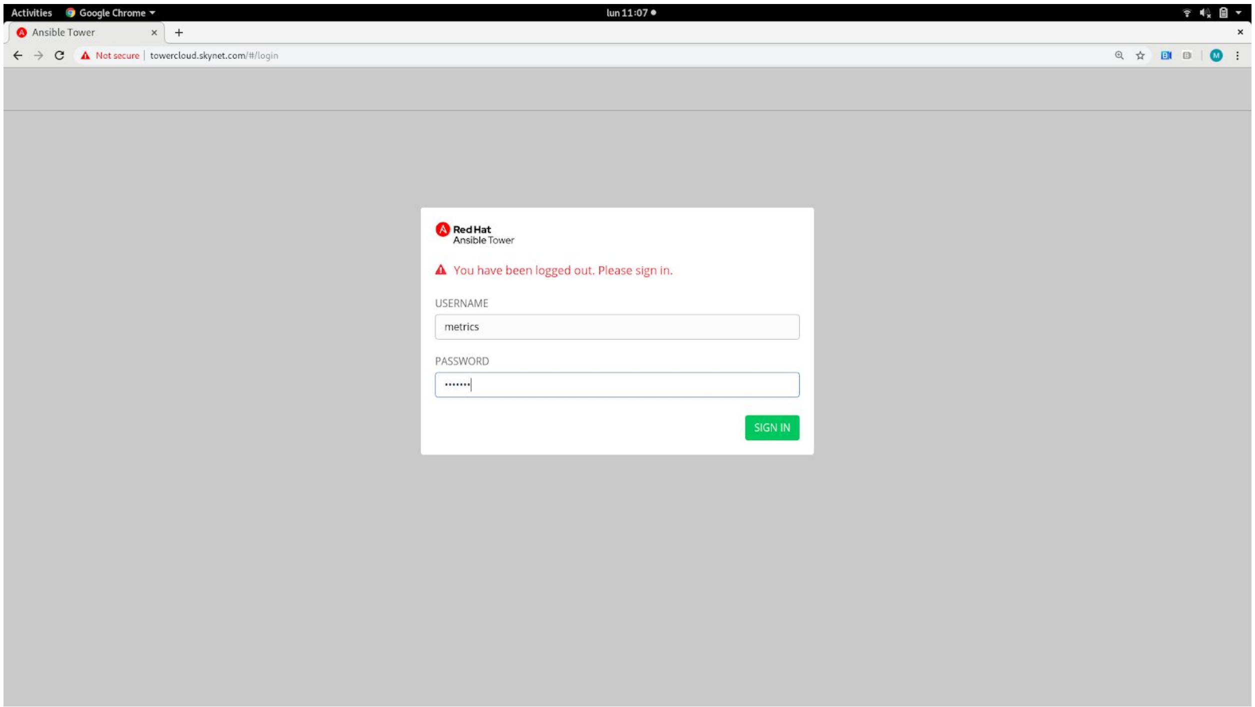
USERS 5

SEARCH | KEY | +

USERNAME	FIRST NAME	LAST NAME	ACTIONS
admin			
apiroot			
apiuser			
azure			
azureoperator			

And finally, we will create a *Personal OAuth2 Token* [4] for the *metrics* user. This can be done either via the Ansible Tower API or the GUI.

In the following example, we will use the GUI to create the token. First, we need to log in with 'metrics' user that we created earlier.

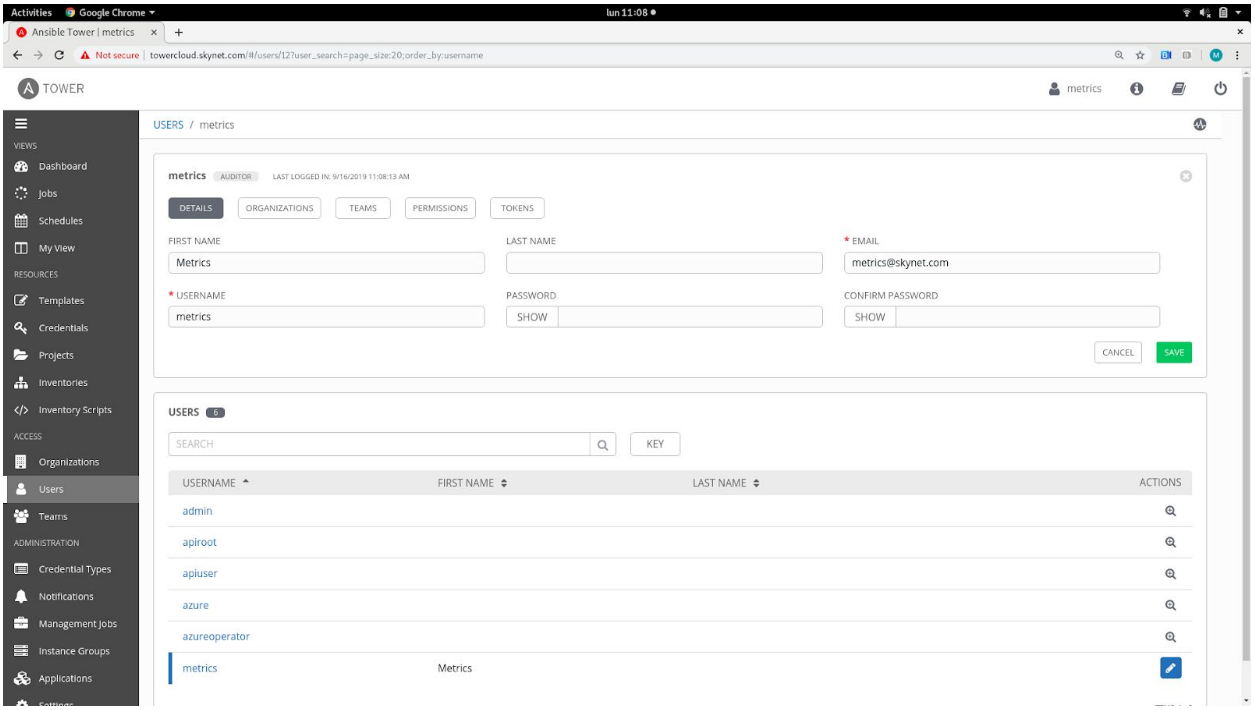


You have been logged out. Please sign in.

USERNAME
metrics

PASSWORD
•••••••

Then, from the Users options on the left hand side menu, we access the list of the users, click on metrics, and click on the edit icon on the right.



metrics AUDITOR LAST LOGGED IN: 9/16/2019 11:08:13 AM

FIRST NAME: Metrics

LAST NAME:

* EMAIL: metrics@skynet.com

* USERNAME: metrics

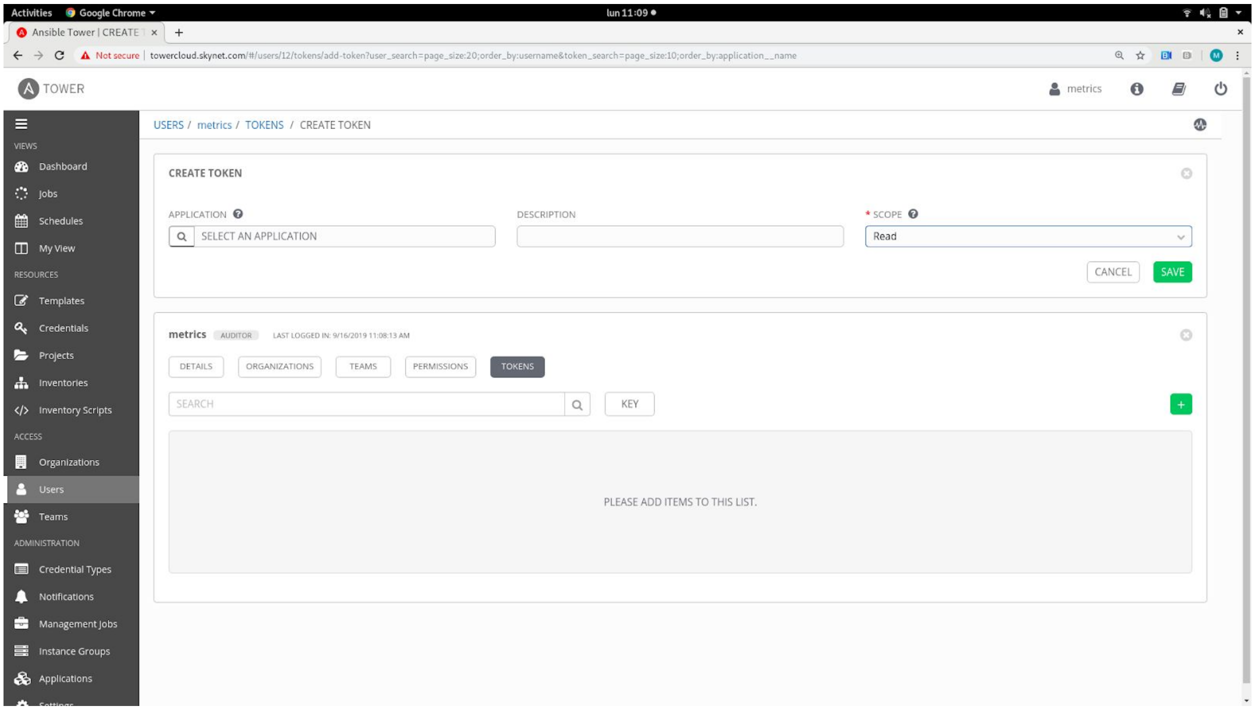
PASSWORD: SHOW

CONFIRM PASSWORD: SHOW

USERS 6

USERNAME	FIRST NAME	LAST NAME	ACTIONS
admin			
apiroot			
apiuser			
azure			
azureoperator			
metrics	Metrics		

Select the *Token* button and add a new Token (+) and set the *Scope* option of *Read*.



CREATE TOKEN

APPLICATION: SELECT AN APPLICATION

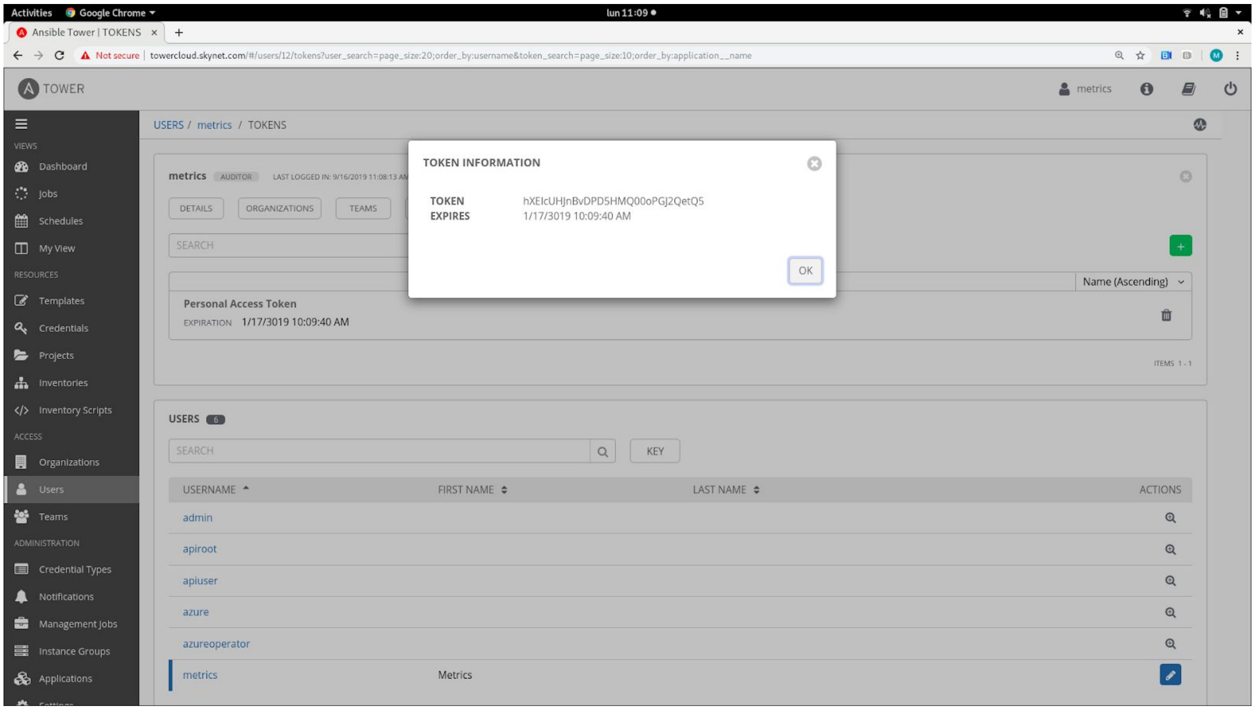
DESCRIPTION:

* SCOPE: Read

metrics AUDITOR LAST LOGGED IN: 9/16/2019 11:08:13 AM

PLEASE ADD ITEMS TO THIS LIST.

Click Save, and don’t forget to save the generated OAUTH2 token :)



TOKEN INFORMATION

TOKEN	hXEIcUHJnBvDPD5HMQ00oPGJ2QetQ5
EXPIRES	1/17/3019 10:09:40 AM

Personal Access Token
EXPIRATION 1/17/3019 10:09:40 AM

USERS 6

USERNAME	FIRST NAME	LAST NAME	ACTIONS
admin			
apiroot			
apiuser			
azure			
azureoperator			
metrics	Metrics		

Now, we should have access to the metrics. Let's run the curl command to test.

```
[root@prometheus ~]# curl -sk "https://towercloud.skynet.com/api/v2/metrics/" -H
"Authorization: Bearer hXEIcUHJnBvDPD5HMQ00oPGJ2QetQ5"
# HELP awx_system_info AWX System Information
# TYPE awx_system_info gauge
awx_system_info{ansible_version="2.8.4",external_logger_enabled="False",external_logger_type
="None",insights_analytics="False",install_uuid="9360e3ed-9e9b-4edb-8b42-b82971fe5aff",licen
se_expiry="81177371",license_type="enterprise",pendo_tracking="detailed",tower_url_base="htt
ps://towercloud.skynet.com",tower_version="3.5.2"} 1.0
# HELP awx_organizations_total Number of organizations
# TYPE awx_organizations_total gauge
awx_organizations_total 4.0
# HELP awx_users_total Number of users
# TYPE awx_users_total gauge
awx_users_total 6.0
# HELP awx_teams_total Number of teams
… … …
# HELP awx_license_instance_free Number of remaining managed hosts provided by your license
# TYPE awx_license_instance_free gauge
awx_license_instance_free 498.0
```

Yeah!!! It worked!!! We successfully accessed the Ansible Tower metrics using the 'metrics' user we created in the previous steps :)

The next step is to integrate the metrics we are collecting with our system monitoring tool. In this example, we will use *Prometheus* and *Grafana*. [5]

The assumption is that Prometheus and Grafana have already been installed and configured. As this installation is beyond the scope of this article [6]

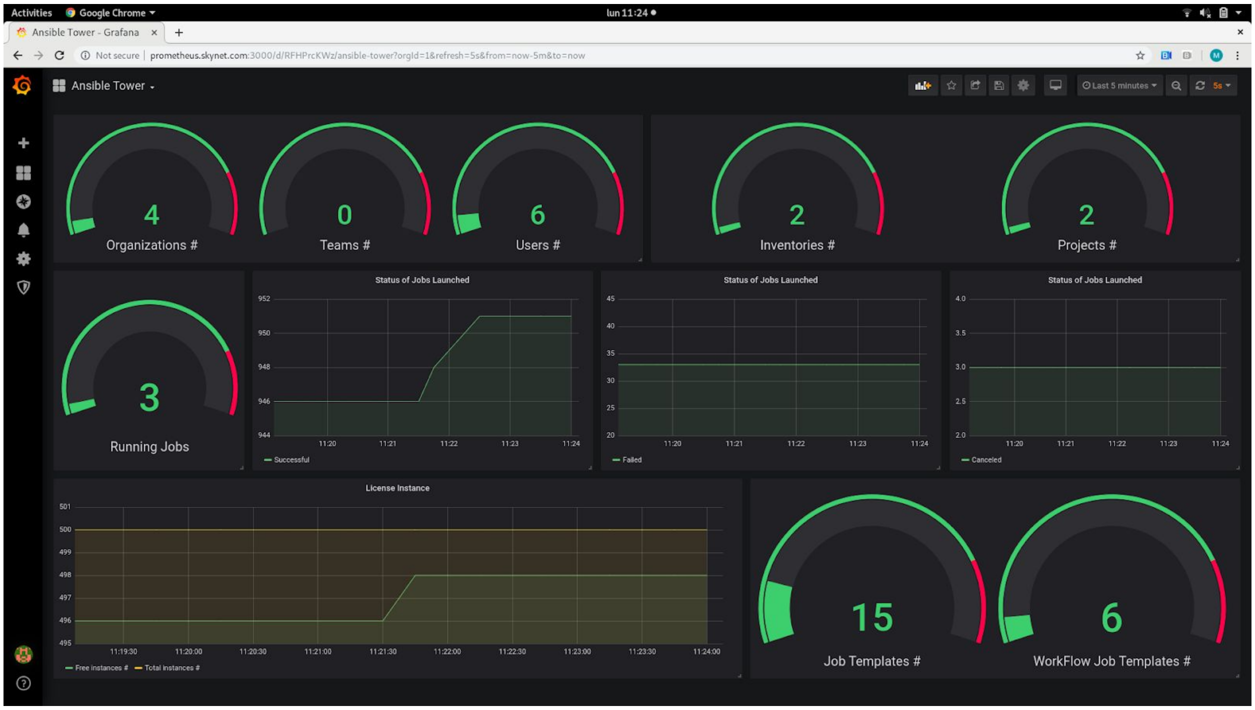
First, we configure Prometheus by editing the /etc/prometheus/prometheus.yml file by adding the following to the *scrape_configs* section:

- `‘Ansible_Tower’` is the name of the job template. You can enter any string value in this field.
- `bearer_token` is the *Personal OAuth2 Token* generated for the *metrics* user in the previous steps.
- `towercloud.skynet.com` is the URL of Ansible Tower.

```
- job_name: 'Ansible_Tower'
   tls_config:
       insecure_skip_verify: True
   metrics_path: /api/v2/metrics/
   scrape_interval: 5s
   scheme: https
   bearer_token: hXEIcUHJnBvDPD5HMQ00oPGJ2QetQ5
   static_configs:
       - targets:
           - towercloud.skynet.com
```

After adding the details, remember to restart/reload the Prometheus service ;)

We are now able to view the data both directly in Prometheus and also visualise it on Grafana dashboard.



What we can see in the Grafana dashboards is as follows:

- Number of Organizations, Teams, Users, Inventories and Projects
- Running jobs
- The Jobs status outputs (Successful, Failed or Cancelled).
- The total number of nodes the Ansible Tower is licensed for
- Number of job templates and workflows

I hope this guide helps you integrate your Ansible AutomationPlatform with the monitoring tools. Don't forget to share your Grafana dashboards with the community! The dashboard I created can be found here [7].

That's all folks!!!

Useful links:

[1] Ansible Tower Release Notes. https://docs.ansible.com/ansible-tower/latest/html/release-notes/relnotes.html#ansible-tower-version-3-5-0

[2] Summary of authentication methods for Red Hat Ansible Tower. https://www.ansible.com/blog/summary-of-authentication-methods-in-red-hat-ansible-tower

[3] Ansible Tower User Guide. Users. https://docs.ansible.com/ansible-tower/latest/html/userguide/users.html

[4] Ansible Tower Administration Guide. Token-Based Authentication. https://docs.ansible.com/ansible-tower/latest/html/userguide/users.html

[5] Ansible Tower Administration Guide. Metrics. https://docs.ansible.com/ansible-tower/latest/html/administration/metrics.html

[6] First steps with Prometheus. https://prometheus.io/docs/introduction/first_steps/

[7] Grafana Dashboard used in this post. https://github.com/MoyRivera/Grafana_Dashboard/blob/master/Ansible%20Tower-1568243189650.json