Ansible Workshop - Exercises

Projects

Use your Ansible skills to complete a couple of small projects.

Project - Linux automation

To further enhance your Ansible skills, let's deploy the monitoring tool *Grafana* to one of the nodes in the demo environment.

Objective

Create an Ansible project *from scratch* and automate some basic linux configurations.

Guide

Step 1 - Prepare project

Create a new project folder in your home directory:

```
[student@ansible-1 ~]$ mkdir grafana-deployment
```

Create an inventory file with a *grafana* group definition. You will deploy Grafana to one of the nodes in the lab environment. Copy the *node2* configuration from the default inventory file to your *grafana* group.
Create a small Ansible configuration file (`ansible.cfg`) and instruct Ansible to always use the inventory you just created.

For example, you may check your inventory with the `ansible-inventory` CLI utility. In this case, the host has an alias of *grafana-instance1* which is part of a group *grafana*:

```
[student@ansible-1 ~]$ ansible-inventory --graph --vars
@all:
  |--@ungrouped:
  |--@grafana:
  |  |--grafana-instance1
  |  |  |--{ansible_host = node2.example.com}
```

Hint

As you can see above, no inventory was provided in the CLI call (e.g. with `-i inventory`), but the correct inventory is used.

Achieve the following tasks:

- Inventory file created
- Configuration file created which sets the correct inventory source

Step 2 - Install Grafana

The Grafana package comes from a dedicated repository, you'll need to enable it for the *yum* package manager on *node2*. Use the following file and copy it to `/etc/yum.repos.d/grafana.repo` with an Ansible task:

```
[grafana]
name=grafana
baseurl=https://packages.grafana.com/oss/rpm
repo_gpgcheck=1
enabled=1
gpgcheck=1
gpgkey=https://packages.grafana.com/gpg.key
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
```

The next task should install the `grafana` package. Another task is needed to start (and enable) the `grafana-server` service.

Achieve the following tasks:

- Running Grafana instance on node2
- Grafana service running and enabled at startup

Ensure that Grafana is running with an ad hoc command:

```
[student@ansible-1 grafana-deployment]$ ansible grafana -a "systemctl status grafana-server"
node2 | CHANGED | rc=0 >>
● grafana-server.service - Grafana instance
   Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; enabled; vendor preset:
disabled)
   Active: active (running) since Sun 2022-04-17 10:00:35 UTC; 2min 44s ago
     Docs: http://docs.grafana.org
 Main PID: 20887 (grafana-server)
    Tasks: 7 (limit: 4579)
   Memory: 97.7M
   CGroup: /system.slice/grafana-server.service
           └─20887 /usr/sbin/grafana-server --config=/etc/grafana/grafana.ini --
pidfile=/var/run/grafana/grafana-server.pid --packaging=rpm
cfg:default.paths.logs=/var/log/grafana cfg:default.paths.data=/var/lib/grafana
cfg:default.paths.plugins=/var/lib/grafana/plugins
cfg:default.paths.provisioning=/etc/grafana/provisioning
```

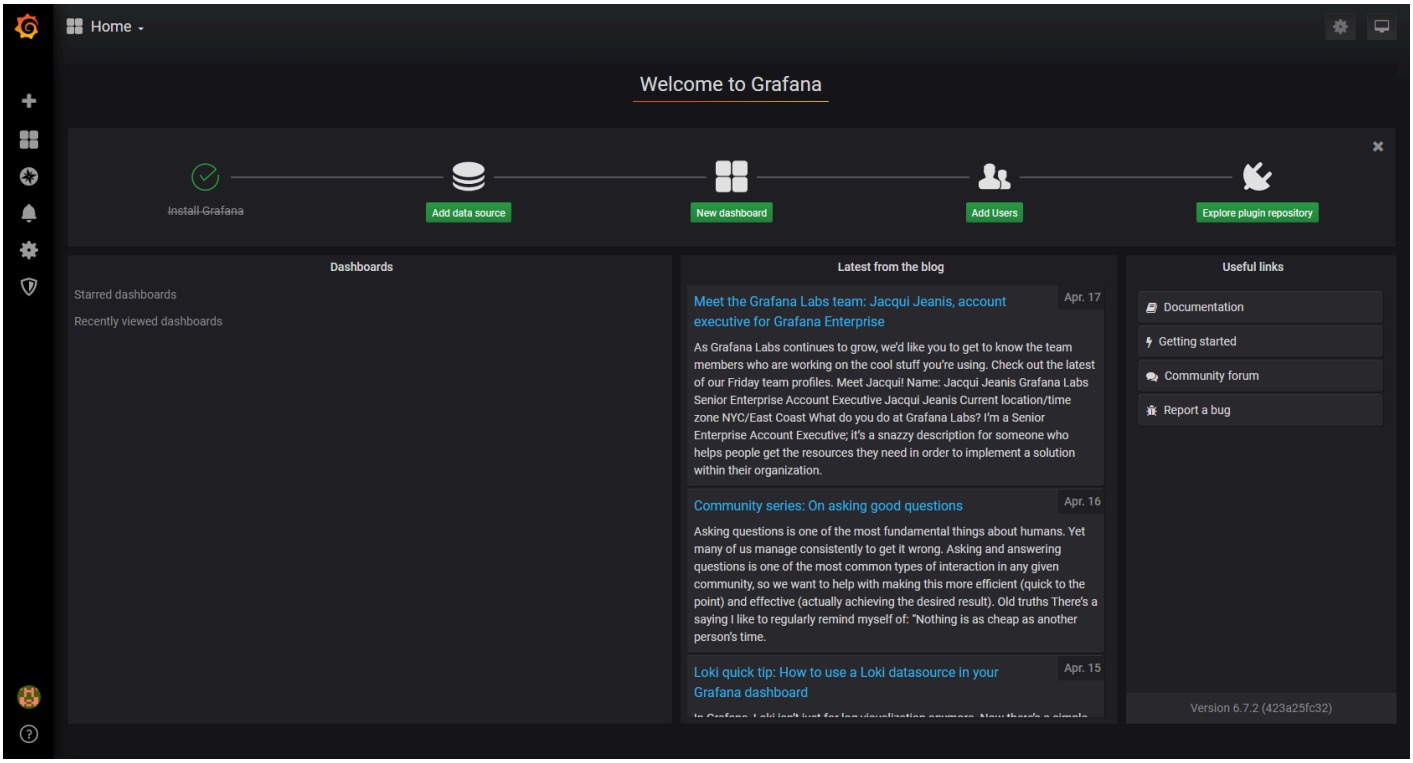
Accessing the Grafana UI from the browser currently fails with a timeout, use the **public** IP address of your *node2* (the one from your inventory) and port 3000 (this is the default port for Grafana). We will fix this in the next step.

Step 3 - Configure Grafana

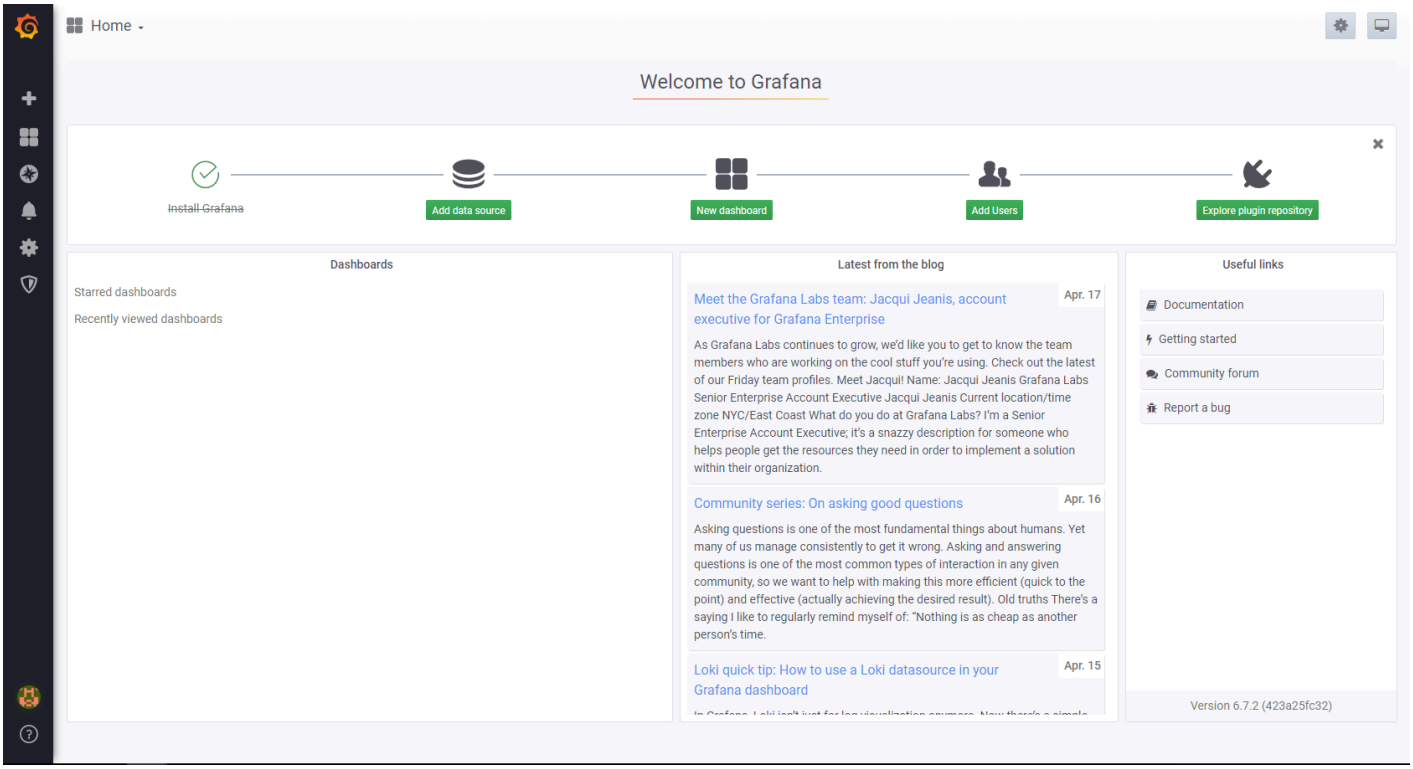
Currently, you are not able to access the Grafana UI, using the **public** IP address of your *node2* and the Grafana default port of 3000, you will get a timeout.
The lab environment only allows access to Port 80 and 8080, yesterday you started an Apache webserver on these ports with Ansible. You'll have to configure Grafana to start on Port 8080 to able to access the UI.

> **Warning**
>
> There should be no running Apache webserver on *node2*, if otherwise, you'll need to stop *httpd* on *node2*! If the port is occupied, Grafana can not be started!
> You could (and should!) ensure a stopped Apache easily with an Ansible task...

By default, Grafana uses a black background. You will adjust the Grafana configuration with Ansible to show the Grafana UI with white background. You will change the look from this...



...to this...



The configuration for Grafana is stored in `/etc/grafana/grafana.ini`. You need to adjust the theme configuration in the *users* section, as well as the *http_port* in the *server* section. Take a look at the Grafana documentation on how to change the parameters.
Naturally, you should achieve this with Ansible! Find an appropriate module (there is more than one way to achieve the solution...) and adjust the Grafana configuration file.

> **Tip**
>
> Configuration changes require a service restart!

After adjusting the configuration, try to access the Grafana UI again. Use the hostname (or public IP address) of your *node2* and use Port 8080 this time.

Success

Unfortunately, the UI currently can't be viewed directly in the Red Hat Demo environment!
If you are in a local environment, you can use the default login credentials *admin:admin*, you can skip the password change request.
You can check if the UI is available by using the curl request `curl -L node2:8080`

Example output

```
[student@ansible-1 ansible-files]$ curl -L node2:8080
<!DOCTYPE html>
<html lang="en">
  <head>
    <script nonce="">

      !(function () {
        if ('PerformanceLongTaskTiming' in window) {
          var g = (window.__tti = { e: [] });
          g.o = new PerformanceObserver(function (l) {
            g.e = g.e.concat(l.getEntries());
          });
          g.o.observe({ entryTypes: ['longtask'] });
        }
      })();
    </script>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
    <meta name="viewport" content="width=device-width" />
    <meta name="theme-color" content="#000" />

    <title>Grafana</title>

    <base href="/" />

    <link
      rel="preload"
      href="public/fonts/roboto/RxZJdnzeo3R5zSexge8UUVtXRa8TVwTICgirnJhmVJw.woff2"
      as="font"
      crossorigin
    />

    <link rel="icon" type="image/png" href="public/img/fav32.png" />
    <link rel="apple-touch-icon" sizes="180x180" href="public/img/apple-touch-icon.png" />
    <link rel="mask-icon" href="public/img/grafana_mask_icon.svg" color="#F05A28" />
    <link rel="stylesheet" href="public/build/grafana.dark.3b87c7ad03e52dfc5e30.css" />

    <script nonce="">
      performance.mark('frontend_boot_css_time_seconds');
    </script>

    <meta name="apple-mobile-web-app-capable" content="yes" />
    <meta name="apple-mobile-web-app-status-bar-style" content="black" />
    <meta name="msapplication-TileColor" content="#2b5797" />
    <meta name="msapplication-config" content="public/img/browserconfig.xml" />
  </head>

  <body class="theme-light app-grafana">
    <style>
      .preloader {
        height: 100%;
        flex-direction: column;
        display: flex;
        justify-content: center;
        align-items: center;
      }
  ...<cut for readability>...
```

You can use the following playbook the check the current theme setting, create a new file, paste to content and run it:

```
- name: Test Grafana theme setting
  hosts: node2
  tasks:
    - name: Get Grafana UI content
      ansible.builtin.uri:
        url: http://node2:8080
        return_content: true
      register: grafana_ui_content

    - name: Output current theme setting
      ansible.builtin.debug:
        msg: "HTML body returns '{{ grafana_ui_content.content | replace('\n', '') |
regex_replace('^(.*body class=\\\")(.*)( app-grafana.*)', '\\2') }}' as the current color setting."
```

Achieve the following tasks:

- Accessible Grafana UI on port 8080
- Grafana UI in `light` theme
- Bonus: Can you manage to control the look of Grafana by just switching a variable?

Step 4 - Re-format project to role structure

All Ansible projects should use the role structure, if your project does not already uses it, now is the time to rearrange your content. Create a `roles` folder and an appropriately named sub-folder for the grafana deployment with all necessary folder and files.
Change your playbook to use your role, e.g.:

```
---
# This is the main Playbook for the 'Grafana Deployment' Project

- name: Deploy Grafana instance
  hosts: grafana
  roles:
    - grafana
```

Make sure everything works by executing your playbook again, you should not see any changes, all tasks should return a green "Ok" status.

Achieve the following tasks:

- Project uses Ansible role structure
- Playbook references role

Step 5 - Bonus: Upload project to Github

Create a new project in your personal Github account and commit your Ansible project.

Step 6 - Bonus: Run your project within AAP

Create a new project in AAP, reference your Grafana project from Github as the code source. Create a template and run your playbook.