Workshop Guide: Using the FBConfigDrift.ps1 Script

Welcome to the workshop on using the `FBConfigDrift.ps1` script! This script focuses around FlashBlade configuration drive management using the API directly in PowerShell.

This guide will walk you through understanding the configuration file, adding additional arrays, modifying the banner property, and changing other properties. By the end of this workshop, you will have a solid understanding of how to use this script to manage configuration drift in your Pure Storage FlashBlade environment.

We even have a few challenges for you to enhance the solution further.

Before you start

You will need to login to the FlashBlade and generate a new API token for the pureuser account, and paste it into your FBArrayConfig.json

1. Understanding the Configuration File

The configuration file (`FBarrayconfig.json`) is a JSON file that holds the configuration settings for your FlashBlades and sites. Here's a breakdown of its structure:

```json
{
    "flashblades": [
        {
            "name": "FlashBlade1",
            "APIToken": "T-4a4c121c-c920-4223-ae0f-321a436ba747",
            "Site": "FBSiteA"
        }
    ],
    "sites": [
        {
            "name": "FBSiteA",
            "settings": [
                {
                    "Name": "BannerPath",
                    "Value": ".\\banners\\standardbanner.txt"
                },
                {
                    "Name": "GUIIdleTimeoutMinutes",
                    "Value": "30"
                },
                {
                    "Name": "NTPServersCommaDelmited",
                    "Value":
"time1.purestorage.com,time2.purestorage.com,time3.purestorage.com"
                }
            ]
        },
```

```json
        {
            "name": "FBSiteB",
            "settings": [
                {
                    "Name": "BannerPath",
                    "Value": ".\\banners\\standardBanner.txt"
                },
                {
                    "Name": "GUIIdleTimeout",
                    "Value": "25"
                },
                {
                    "Name": "NTPServersCommaDelmited",
                    "Value":
"SiteB1.purestorage.com,SiteB2.purestorage.com,SiteB3.purestorage.com"
                }
            ]
        }
    ]
}
```

- **Arrays Section:** This section lists all the FlashBlades. Each array has a name, APIToken, and associated site.

- **Sites Section:** This section lists all the sites and their settings. Each site has a name and a list of settings. Each setting has a name and a value.

2. Adding Additional Arrays

To add an additional array to the configuration, follow these steps:

1. **Identify the new array details** (name, APIToken, site).

2. **Add the new array to the arrays section** of the JSON file.

For example, if you want to add a new array named `FlashBlade2`, your updated `flashblades` section would look like this:

```json
"flashblades": [
    {
        "name": "FlashBlade1",
        "APIToken": "TokenforourServiceUser",
        "Site": "FBSiteA"
    },
    {
        "name": "FlashBlade2",
        "APIToken": "TokenforourServiceUser",
        "Site": "FBSiteB"
    }
```

```
    ]
```

3. Modifying the Banner Property

To modify the banner property, you need to update the `BannerPath` value in the `sites` section.

1. **Identify the site you want to modify**.

2. **Update the `BannerPath` value** to point to the new banner file.

For example, to change the banner for `FBSiteA` to a new file named `newbanner.txt`, update the `BannerPath` setting:

```
"settings": [
    {
        "Name": "BannerPath",
        "Value": ".\\banners\\newbanner.txt"
    },
    {
        "Name": "GUIIdleTimeoutMinutes",
        "Value": "30"
    },
    {
        "Name": "NTPServersCommaDelmited",
        "Value":
"time1.purestorage.com,time2.purestorage.com,time3.purestorage.com"
    }
]
```

4. Changing Other Properties

Let's walk through changing some other properties like `GUIIdleTimeoutMinutes` and `NTPServersCommaDelmited`.

Changing `GUIIdleTimeoutMinutes`

1. **Locate the site** in the JSON file.

2. **Update the `GUIIdleTimeoutMinutes` value**.

For example, to change the `GUIIdleTimeoutMinutes` for `FBSiteA` to 45 minutes:

```
"settings": [
    {
```

```
            "Name": "GUIIdleTimeoutMinutes",
            "Value": "45"
        }
    ]
```

Changing `NTPServersCommaDelmited`

1. **Locate the site** in the JSON file. This is a comma separated field

2. **Update the `NTPServersCommaDelmited` value**.

For example, to change the NTP servers for `FBSiteA`:

```
"settings": [
    {
        "Name": "NTPServersCommaDelmited",
        "Value":
"time4.purestorage.com,time5.purestorage.com,time6.purestorage.com"
    }
]
```

5. Running the Script

You can run the script in two modes: `plan` and `apply`.

- **Plan Mode:** This mode will show pending changes without applying them.

- **Apply Mode:** This mode will apply the changes.

Example command to run in plan mode:

```
./FBConfigDrift.ps1 –mode "plan"
```

Example command to run in apply mode and open an incident if a drift is detected:

```
./FBConfigDrift.ps1 –mode "apply" –openincident $true
```

Things to observe in the solution

- We are operating with the input in this example of stored Hash tables in the config file. This gives us a key value/pair type format for storing and replaying the data.

  - The benefit of this design, is it allows for ease of adding additional configuration, with minimal changes to the script. We should only need to modify the `Set-FBconfig` and our JSON file to add new checks, which is the goal of this simple workshop.

  - The downside to this approach is, things such as SNMP server, SMTP mail, etc sometimes have multiple values needed. It is recommended, in this format, to make those fields comma delimited to hold multiple properties.

- NTPServersCommaDelmited shows an example of how to import CSV format and comparing two arrays of strings. It is interesting for this type of field, because we want to compare all the various properties, regardless of the order. This comparison example will not care about order of the defined elements in the config vs the array.

Challenge

Add a new property to our JSON file and a check to add into `Set-FBconfig`

Simple Examples

- Set Timezone using get and patch on endpoint `api/2.XX/arrays`

Harder Examples

- Alert Watcher List using get and post on endpoint `/api/2.XX/alert-watchers`
- SMTP Configuration using get and post on endpoint `/api/2.XX/smtp-servers`

Conclusion

In this workshop, you've learned how to understand and modify the configuration file for the `FBConfigDrift.ps1` script. You've seen how to add additional arrays, change the banner property, and modify other settings. You've also learned how to run the script in different modes.

If you have taken the challenges, you have learned how to add a new property and add additional checks to this script

Feel free to experiment with the configuration and script to get more comfortable with its functionality.