AWS/AOP/K8S SSO Cheat Sheet

Document Level Classification

200

Introduction

This document is meant to serve as a reference for commonly used AOP and Kubectl commands. If you are familiar with AOP use the section titled **Users familiar with AOP**. If you are not familiar with AOP and are using it for the first time, use the section titled **First time users of AOP**.

Prerequisites

AOP Installed and Updated to the Latest Version

- If installing for the first time as part of the developer toolchain please refer to the Substrate Prerequisites page for installation instructions for the Local Developer Toolchain. Instructions for MacOS and Windows (Powershell) are provided.
- If Upgrading please refer to Using aop with Substrate.

Kubelogin Installed and Updated to the Latest Version

- If installing for the first time as part of the developer toolchain please refer to the Substrate Prerequisites page for installation instructions for the Local Developer Toolchain. Instructions for MacOS and Windows (Powershell) are provided.
- If Upgrading please refer to Using aop with Substrate.

Command Reference

If you have already used aop shell/kube commands in the past, the commands have been deprecated. Expand the section for **Users Familiar**

with AOP for a cheat sheet that shows how to migrate from the old commands to the new commands.

Command(s)	Usage
`aop aws-sso generate`	Users need to run this the fist time they use the new access pattern or when they get access to a new account/role. It will create a back up of the config and credentials file if they exist. It is the users responsibility to save these backups if they want them. It will then overwrite the config file. It does not create a credentials file. WARNING: This command will backup any existing config or credentials files. It is the user's responsibility to save the backups to a safe location. Future executions of this command will overwrite any existing backups.
`aop aws-sso browser`	This does not open the browser directly into an account. Instead this opens the browser to the AWS SSO user UI where one can then access the target account.
`aws configure list-profiles`	This will list the profiles that the user has access to. On Mac/Unix/Linux systems, users can pipe this into `sort` to get an alphabetically sorted list.

	On Windows systems user can pipe this into `Sort-Object` to get an alphabetically sorted list. The output of the command will provide you with the `<account>-<role>` to be used in the export `AWS_PROFILE=<account>-<role>`
`aws sso login`	The `aws sso login` command only needs to be run once a day. It does not target a specific account or role.

`aop aws-sso generate`
 	
`aws configure list-profiles`
 	
`export AWS_PROFILE=<account>-<role>`
 	
`aws sso login`
	This full example shows the order of operations for generating a list of profiles, listing the available profiles, exporting that profile and logging into aws . The `aop aws-sso generate` only needs to be ran if it is the first time the user is logging in or needs to regenerate the list of available profiles. Exporting the profile is done as needed to change accounts/roles. The `aws sso login` command only needs to be run once a day.

`aop k8s-sso eks-generate`
 	
`kubectl config get-contexts`
 	
`kubectl config use-context <cluster>`
	The example assumes the user already exported the proper AWS_PROFILE. It's recommended that one unsets the environment variable **KUBECONFIG**. The `k8s-sso eks-generate` command will add contexts to the current **KUBECONFIG** file for the clusters in the AWS account you're currently logged into. This is intentional because the authentication to Kubernetes is decoupled from one's AWS identity.

`aop aws-sso update`	Run this command when you're trying to authenticate to relatively new account and getting the following: `Error: There is no account named: dced-dev`

Users familiar with AOP

The following table shows the old version of each command and its new command equivalent. Where an old command has been deprecated a new command is provided.

Old Command	**New Command**	**Usage**
N/A	`aop aws-sso generate`	Users need to run this the fist time they use the new access pattern or when they get access to a new account/role. It will create a back up of the config and credentials file if they exist. It is the users responsibility to save these backups if they want them. It will then overwrite the config file. It does not create a credentials file.
`aop login <account> <role>`	`aop aws-sso browser`	This does not open the browser directly into an account. Instead this opens the browser to the AWS SSO

		user UI where one can then access the target account.
`aop auth <account> <role>`	`aws sso login`	The `aws sso login` command only needs to be run once a day. Exporting the profile is done as needed to change accounts/roles.

`aop shell <account> <role>`	`aws sso login`
	
	`export AWS_PROFILE=<account>-<role>`
		The `aws sso login` command only needs to be run once a day. Exporting the profile is done as needed to change accounts/roles.

N/A	`aws configure list-profiles`	This will list the profiles that the user has access to. On Mac/Unix/Linux systems, users can pipe this into `sort` to get an alphabetically sorted list. On Windows systems user can pipe this into `Sort-Object` to get an alphabetically sorted list.

`aop kube <cluster>`	`aop k8s-sso eks-generate`
	
	`kubectl config get-contexts`
	
	`kubectl config use-context <cluster>`
		It's recommended that one unsets the environment variable **KUBECONFIG**. The `k8s-sso eks-generate` command will add contexts to the current **KUBECONFIG** file for the clusters in the AWS account you're currently

		logged into. This is intentional because the authentication to Kubernetes is decoupled from one's AWS identity.

Using AWS SSO for the First Time

```
❯ $(aop aws-sso creds <account> <role>)
If your browser did not open up automatically go to the following URL:
      https://device.sso.us-east-1.amazonaws.com/?user_code=

    The following user code is provided in case it wasn't automatically


    Please enable JavaScript if you use browser plugins that block JS e

    ******************************************************
    *** Press [ENTER] once you're done in your browser ***
    ******************************************************
```

```
➔ aop aws-sso generate
Made a copy of /Users/firstname.lastname/.aws/config to /Users/firstnam
If you want to keep this backup move /Users/firstname.lastname/.aws/con
The next run of aop aws-sso generate will write over this backup
Authorizing Client...

  If your browser did not open up automatically go to the following URL
    https://device.sso.us-east-1.amazonaws.com/?user_code=<redacted>

  The following user code is provided in case it wasn't automatically f
    <redacted>
```

```
  Please enable JavaScript if you use browser plugins that block JS exe
  Waiting for authorization
Gathering account data, this can take a while...
Wrote config to /Users/firstname.lastname/.aws/config
AWS SSO caches accounts/roles. Please allow up to 30 minutes for a new
```

Listing Profiles

```
➜ aws configure list-profiles | sort
Art-Pipeline-devops
Art-Pipeline-infosec
Art-Pipeline-read-only
ArtStation-infosec
ArtStation-read-only
...
```

Authenticating with AWS

```
➜ aws sso login
Attempting to automatically open the SSO authorization page in your def
If the browser does not open or you wish to use a different device to a

https://device.sso.us-east-1.amazonaws.com/

Then enter the code:

<redacted>
Successfully logged into Start URL: https://epicgames.awsapps.com/start

➜ export AWS_PROFILE=cbfa-live-devops
```

```
➜ AWS_PAGER='' aws sts get-caller-identity
{
    "UserId": "AROAXGZBXF4AU6HNGJPWR:firstname.lastname",
    "Account": "495602577153",
    "Arn": "arn:aws:sts::495602577153:assumed-role/AWSReservedSSO_devop
}
```

Using K8S SSO for the First Time

Note: This example assumes the user is already authenticated to the AWS account with the cluster that is being accessed. Furthermore this example expects the user to have completed the installation of https://github.com/int128/kubelogin as mentioned in the requirements section. K8S SSO can be used independently of AWS SSO. This means if you're using UAM auth to AWS SSO you can still try out K8S SSO.

Set Desired Profile

```
➜ export AWS_PROFILE=cbfa-live-devops
➜ export AWS_REGION=us-east-1
```

Generate configuration for KUBECONFIG

```
➜ aop k8s-sso eks-generate
```

List Clusters

```
➜ kubectl config get-contexts
CURRENT   NAME                          CLUSTER                       AUTHINFO
          cbfa-live-infosec             cbfa-live-infosec             k8s-sso
          cbfa-live-onna                cbfa-live-onna                k8s-sso
          onna-prod-495602577153        onna-prod-495602577153        k8s-sso
```

Switch to the Cluster You are Working With

```
➜ kubectl config use-context cbfa-live-infosec
Switched to context "cbfa-live-infosec".
```

Get a List of Namespaces

```
➜ kubectl get ns
NAME                                  STATUS   AGE
apps                                  Active   460d
beats-engine                          Active   460d
canonical                             Active   460d
datalogger                            Active   460d
default                               Active   2y68d
eks-components-tests                  Active   104d
```

```
eks-components-tests-namespace        Active   104d
epic-system                           Active   2y68d
istio-system                          Active   174d
kube-node-lease                       Active   2y68d
kube-public                           Active   2y68d
```
