Gridspace

Sift AWS Integration Documentation

Version 1.0

July 2016

SIFT AWS INTEGRATION DOCUMENTATION, *VERSION 1.0*

The Gridspace Sift API supports easy integration with Amazon Web Services (AWS). This allows you to run the API on your own AWS EC2 instances, giving you all the features of Sift while maintaining full control over your data.

CHAPTER 1: ECS SETUP

The Sift AWS integration relies on the Amazon EC2 Container Service (ECS). ECS allows you to easily deploy and manage Docker containers on EC2 instances.

1.1 Install ECS CLi

To get started, you will need to install the ECS Command Line Interface. For instructions on how to do so see here:
http://docs.aws.amazon.com/AmazonECS/latest/developerguide/ECS_CLI_installation.html.

1.2 Configure ECS CLI

Once you have installed the ECS CLI, you need to do some quick configuration. The ECS CLI must be linked with an AWS Account, a cluster name, and a region. The cluster name will be used to refer to to the cluster of EC2 instances that the Sift API Containers will run on. After this configuration, any ECS CLI commands issued will pertain to this cluster. The region is the AWS region in which the cluster will be located.

As for specifying the AWS account, there are a couple ways of doing so:

1. Use an AWS CLI Profile

If you have already used the AWS CLI and have a local profile set up, you can just use that profile during configuration. To check if you have an existing profile, see if there is a ~/.aws/credentials file on your machine. If there is, it will contain the current profiles' names followed by their credentials:

```
$ cat ~/.aws/credentials
[default]
aws_access_key_id = ABCDEFGHIJKLMN123456
aws_secret_access_key = A+HPd9gVxbxc622234ccc88SDvlkDFKJlxU123BB
[test]
aws_access_key_id = ABCDEFGFEDCBA1234321
aws_secret_access_key = BBOU123+osDFKJllkjdf+A1234fsd1000SDFBBCD
```

To configure you can then run `ecs-cli configure` as follows:

```
$ ecs-cli configure --cluster my-sift-cluster --region us-west-2 --profile default
INFO[0000] Saved ECS CLI configuration for cluster (my-sift-cluster)
```

This will then create a cluster named `my-sift-cluster` in the `us-west-2` region under the `default` AWS profile.

2. Use AWS access keys directly

If you don’t have an AWS CLI profile set up and don’t want to set one up, you can pass the access keys to your AWS account directly into the `ecs-cli configure` command

```
$ ecs-cli configure --cluster my-sift-cluster --region us-west-2 --access-key
→ABCDEFGHIJKLMN123456 --secret_key A+HPd9gVxbxc622234ccc88SDvlkDFKJlxU123BB
```

CHAPTER 2: ADD AN INSTANCE

Now that the ECS CLI is configured, you can use it to add some EC2 instances to your cluster. To do this, use the ecs-cli upcommand:

```
$ ecs-cli up --keypair my_keypair_name --capability-iam --instance-type t2.large
INFO[0000] Created cluster                                cluster=my-sift-cluster
INFO[0001] Waiting for your cluster resources to be created
INFO[0001] Cloudformation stack status                    stackStatus=CREATE_IN_
→PROGRESS
INFO[0061] Cloudformation stack status stackStatus=CREATE_IN_
→PROGRESS
```

You should substitute the name of your key pair to allow SSH access to your instance. For help on setting up key pairs, see Amazon’s guide. Make sure that the given keypair is tied to the same region specified when runing the `ecs-cli configure` command. The `capability-iam` flag is required to acknowledge that the command may create IAM resources. You may substitute your own EC2 `instance-type` value, but keep in mind that Sift requires at least 8GB of memory tor un on a single instance.For more details on the `ecs-cli` up command, see the developer guide.

After running this command you will have an EC2 instance running in your cluster. Now you need only deploy Sift to the instance.

CHAPTER 3: DEPLOY SIFT

You should have received a `sift-aws.yml` file from Gridspace for deploying Sift. Now all you need to do is run:

```
$ ecs-cli compose --file sift-aws.yml --project-name sift service up
INFO[0000] Using ECS task definition →sift:1
INFO[0000] Created an ECS service
→sift taskDefinition=ecscompose-sift:1
INFO[0000] Updated ECS service successfully →serviceName=ecscompose-service-sift
INFO[0060] Describe ECS Service status →runningCount=0
serviceName=ecscompose-service-sift
INFO[0091] ECS Service has reached a stable state →runningCount=1
serviceName=ecscompose-service-sift
TaskDefinition=ecscompose-
service=ecscompose-service-
desiredCount=1
desiredCount=1
desiredCount=1
```

You can substitute whatever `project-name` you would like.
This command will deploy a Sift container to the instance you created and then ensure that if any failures occur, another container will be started. Behind the scenes this is doing a couple of things. It creates an ECS Task Definition from the `sift-aws.yml` compose file, and then creates a Service out of the task. For more details see the developer guide.