

Configuring DataSunrise Containers with Cloud Formation script on AWS ECS

Instruction Manual

March, 2021

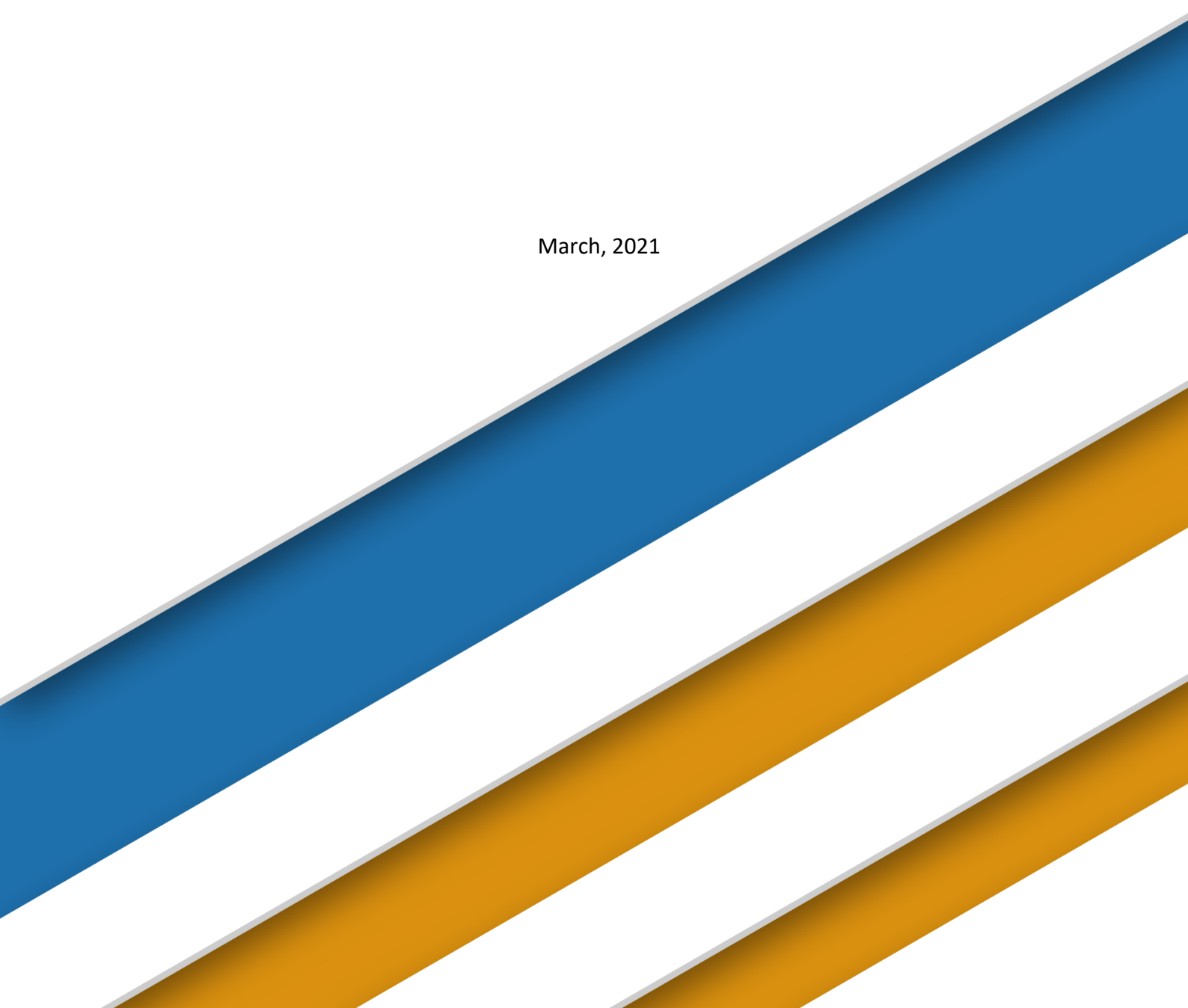
The bottom half of the page is decorated with two thick, parallel diagonal stripes. The top stripe is blue and the bottom stripe is orange, both running from the bottom-left towards the top-right.

Table of Contents

1. Introduction.....	3
1.1 Description and structural scheme of a CloudFormation stack.....	3
2. Prerequisites.....	4
2.1 Public Deployment	5
2.2 Private Deployment.....	5
3. Deploying a CloudFormation stack	5
3.1 Stack Name and Deployment Name	6
3.2 Stack Parameters.....	7
3.3 Finishing the Deployment	10
4. Fault-tolerance and Recovery	12
5. IAM Role and Policies	12
FAQ.....	13

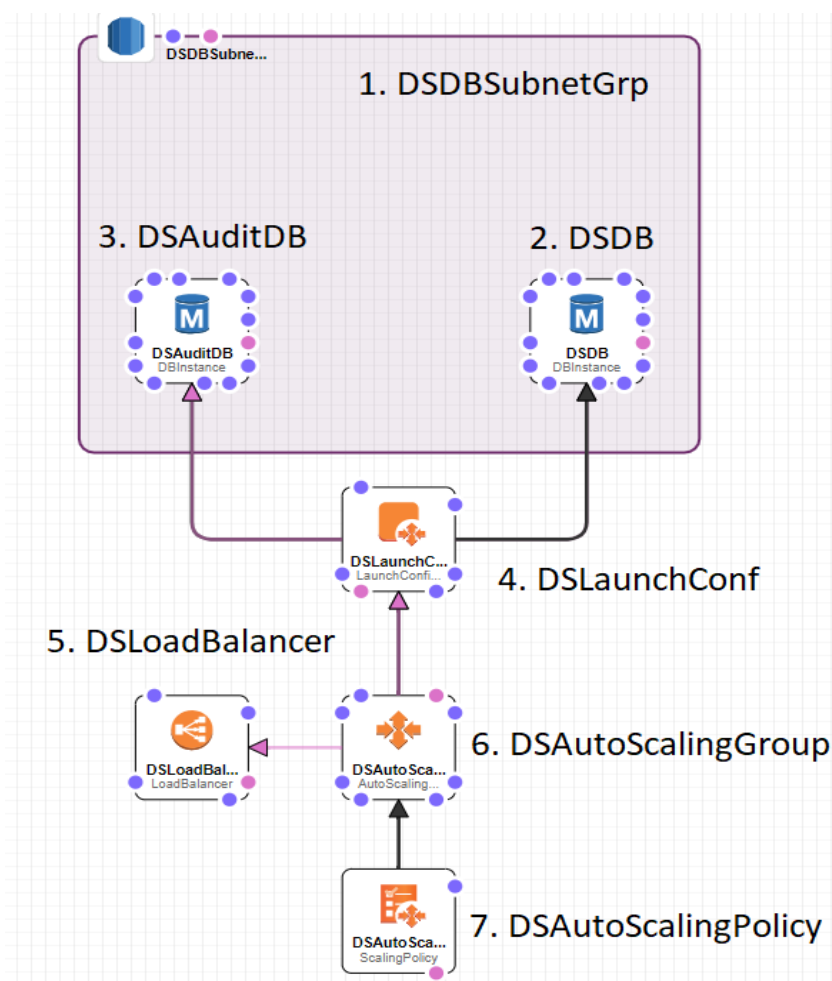
1. Introduction

Manual deployment of a High Availability configuration within any kind of environment presumes considering a lot of different aspects to be kept in mind. Even though a deployment process is successfully finished, there are additional actions to be done for a flawless and stable configuration to be created.

To avoid all of the possible issues and skip manual adjusting, DataSunrise provides a dedicated script for HA environment (infrastructure) deployment within the Amazon Web Service based on Amazon [Cloud Formation](#) Service. The deployment process is automated and does not require manual Amazon Web Service resources creation.

1.1 Description and structural scheme of a CloudFormation stack

The following picture displays the most important objects created by Cloud Formation as they are listed in the script, creation order is parallel, considering objects' dependencies (marked with arrows).



1. DSDBSubnetGrp ([AWS::RDS::DBSubnetGroup](#)) - RDS security group. Restricts access to the Dictionary and Audit Storage databases;
2. DSDictionaryDB ([AWS::RDS::DBInstance](#)) - RDS database instance used to store DataSunrise settings (Dictionary);
3. DSAuditDB ([AWS::RDS::DBInstance](#)) - RDS database instance used to store DataSunrise's audit journal and other journals (Audit Storage);
4. Cluster ([AWS::ECS::Cluster](#)) – ECS Cluster that will group up DataSunrise containers.
5. Service ([AWS::ECS::Service](#)) – Service that will be used within the Cluster;
6. TaskDefinition ([AWS::ECS::TaskDefinition](#)) – Task Definition;
7. LoadBalancer ([AWS::ElasticLoadBalancingV2::LoadBalancer](#)) – Load Balancer for accessing DataSunrise WebUI.
8. AutoScalingTarget ([AWS::ApplicationAutoScaling::ScalableTarget](#)) - Auto scaling policy, defines the way of deployment of new containers.

Note: There are different dependent resources that will be created as well. Describing them here will not bring any useful information.

2. Prerequisites

There are some entities that should exist in your AWS before you start the deployment process. Here is the list of required items:

- Amazon Virtual Private Cloud (VPC) that will be used as the environment where the CloudFormation stack will be deployed;
- Subnets within the VPC that you will designate for your DataSunrise cluster running in the specified VPC;
- Certificate that will be used in the Load Balancer
- Subnets to be used should have the following resource endpoints attached:
 - ECS Endpoint: required for containers accessing ECS services;
 - Secrets Manager Endpoint: required for obtaining DataSunrise administrator password from the Secrets Manager;
 - Cloud Formation Endpoint: required for Cloud Initialization;
 - Cloud Watch Endpoint: required for Cloud Watch synchronization and DataSunrise Suite Log Files upload.
 - NAT Gateway is required in case DataSunrise is deployed in a private subnet and the Hourly Billing license type is used.

Important: The endpoints presented in the list should have private DNS names enabled.

2.1 Public Deployment

To publicly deploy a Cloud Formation stack, you will need to choose two public subnets from the same VPC but from different Availability Zones. An Internet Gateway attached to the VPC is also required.

For each subnet, you must verify that its route table (could be one shared route table or one per subnet) defines a route pointing the destination "0.0.0.0/0" to the Internet Gateway.

When deploying a Cloud Formation stack, you will need to choose Elastic load balancer working mode — select "internet-facing".

2.2 Private Deployment

To privately deploy a Cloud Formation stack, you will need to choose two private subnets from the same VPC but from different Availability Zones. A NAT Gateway attached to the VPC is also required.

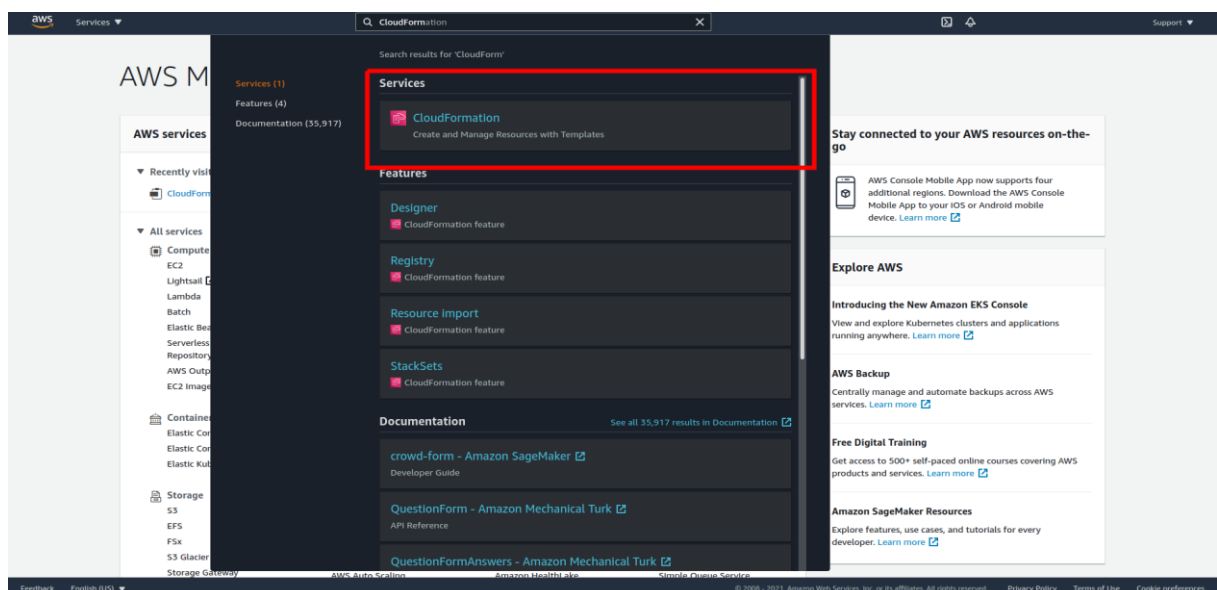
For each subnet, you should verify that its route table (one shared route table or one per subnet) defines a route pointing the destination "0.0.0.0/0" to the NAT Gateway.

When deploying a Cloud Formation stack, you will need to use the "internal" Elastic load balancer mode.

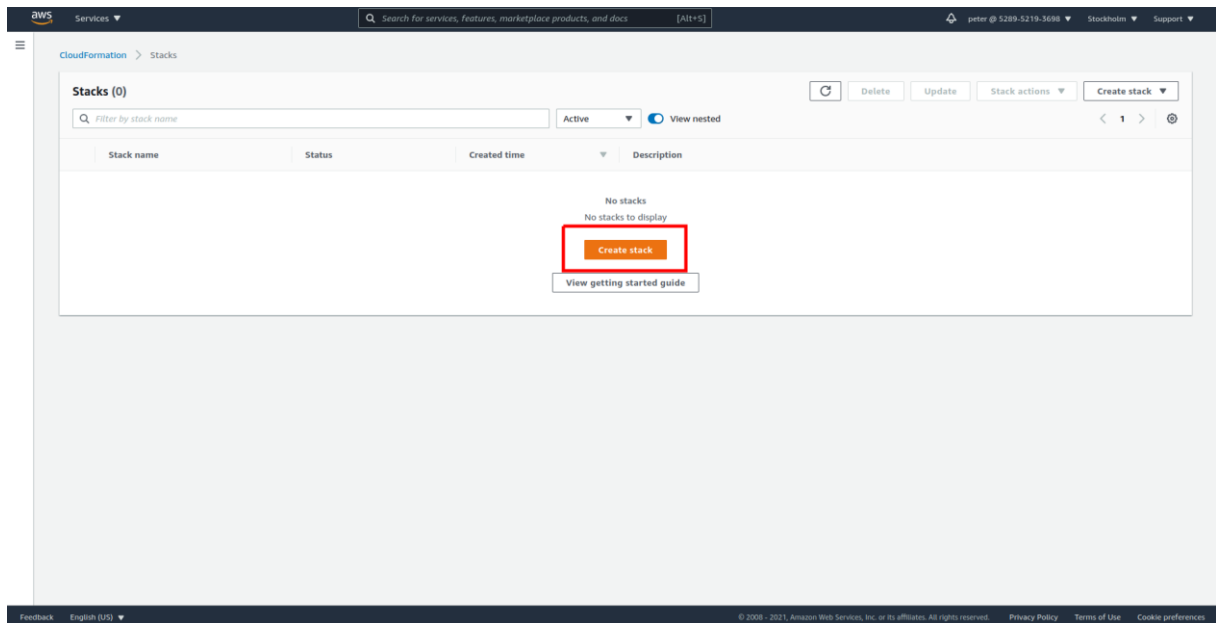
You can also use HTTP Proxy in case NAT Gateway attachment is not possible.

3. Deploying a CloudFormation stack

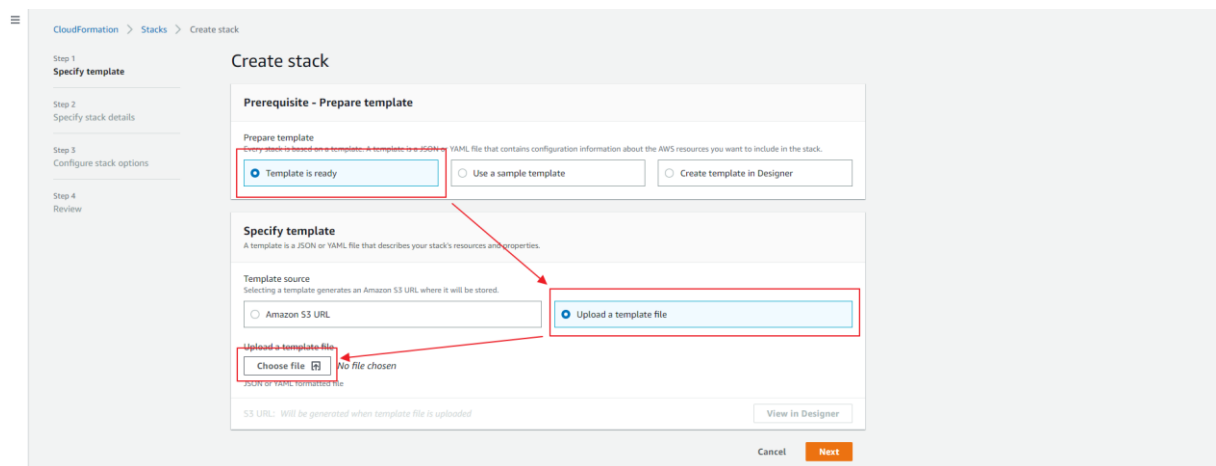
To deploy DataSunrise in HA configuration, navigate to the **Cloud Formation** subsection of AWS Console services.



Click **Create stack**:



Select your Cloud Formation script and click **Next**:



3.1 Stack Name and Deployment Name

Click **Next** and you will be directed to the CloudFormation script's settings. You will see different fields. Some of them are already filled in with default values and some of them are empty. In this instruction, we will dwell on each field.

The first thing that should be done during the deployment process is naming the stack:

Stack name

Stack name

Enter a stack name

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

You can use any name here. This name will be used only to identify the stack during the deployment process and for you to differ it from other deployed stacks. This name will also be concatenated with resource names.

3.2 Stack Parameters

User Location CIDR:

Network Configuration

UserLocationCIDR

0.0.0.0/0

- **UserLocationCIDR** – the address from which a user can access the DataSunrise Console and Proxy Port.
- **Example:** 172.32.0.0/16 to be able to connect to the DataSunrise Proxy or Console from the default VPC IP-range.

Dictionary & Audit Database Configuration

DictionaryType

Postgres

DictionaryDBClass

The database instance class that allocates the computational, network, and memory capacity required by planned workload of this database instance.

db.t3.medium

DictionaryDBStorageSize

The size of the database (Gb), minimum restriction by AWS is 20GB

20

DBDictName

Name of the database to store DataSunrise configuration. Must begin with a letter and contain only alphanumeric characters.

dsdictionary

MultiAzDictionary

Dictionary RDS Multi-AZ

true

AuditType

Postgres

AuditDBClass

The database instance class that allocates the computational, network, and memory capacity required by planned workload of this database instance.

4xlarge

AuditDBStorageSize

The size of the database (Gb), minimum restriction by AWS is 20GB

200

DBAuditName

Name of the database to store DataSunrise audit journal. Must begin with a letter and contain only alphanumeric characters.

dsaudit

MultiAzAudit

Make Audit RDS Multi-AZ

true

DBUser

The database administrator account username. Must begin with a letter and contain only alphanumeric characters.

dsuser

DBPassword

The database administrator account password.

SubnetsConfig

Dictionary and Audit subnets. Must be a part of mentioned VPC. Please be sure that you select at least two subnets.

- **DictionaryType** – choose the type of Dictionary Database among 3 available values that suits your requirements.
- **DictionaryDBClass** – specify the type of the machine the Dictionary Storage will be deployed on.
- **DictionaryDBStorageSize** – specify the storage size of the Dictionary database.
- **DBDictName** - specify the name of internal PostgreSQL RDS database. It will be used to establish a connection between your DataSunrise and the Dictionary.
- **MultiAZDictionary** – choose true in case it is required dictionary db to be available from multiple Availability Zones.

- **AuditType** – choose the type of Audit Database among 3 available values that suits your requirements.
- **AuditDBClass** – specify the type of the machine the Audit Storage will be deployed on.
- **AuditDBStorageSize** – specify the storage size of the Audit database.
- **DBAuditName** – specify the name of internal PostgreSQL RDS database. It will be used to establish a connection between your DataSunrise and the Audit Storage.
- **MultiAZAudit** – choose true in case it is required audit db to be available from multiple Availability Zones.
- **DBUser** – specify the name of the user that will be used to access the Dictionary and Audit databases.
- **DBPassword** – specify the password that will be used along with the DBUser as credentials
- **SubnetsConfig** – specify subnets that will be used for the Dictionary and Audit databases. In case MultiAZ for the dictionary or audit are used – minimum 2 subnets will be required.

Certificate

ContainerCpu

ContainerMemory

- "1GB, 2GB, 3GB, 4GB - Available cpu values: 512 (.5 vCPU)" - "2GB, 3GB, 4GB, 5GB, 6GB, 7GB, 8GB - Available cpu values: 1024 (1 vCPU)" - "Between 4GB and 16GB in 1GB increments - Available cpu values: 2048 (2 vCPU)" - "Between 8GB and 30GB in 1GB increments - Available cpu values: 4096 (4 vCPU)"

ContainerPort

ContainerProxyPort

If container will be used as proxy - specify a port number that will be used for proxy.

ContainersCount

Image

Subnets

VPC

VpcCidr

- **Certificate** – specify the EXISTING certificate ARN that will be used in the Load Balancer
- **ContainerCpu** - the number of cpu units used by the task (container). Use one of the following values, which determines your range of valid values for the memory parameter:
 - 256 (.25 vCPU) - Available memory values: 512 (0.5 GB), 1024 (1 GB), 2048 (2 GB)
 - 512 (.5 vCPU) - Available memory values: 1024 (1 GB), 2048 (2 GB), 3072 (3 GB), 4096 (4 GB)
 - 1024 (1 vCPU) - Available memory values: 2048 (2 GB), 3072 (3 GB), 4096 (4 GB), 5120 (5 GB), 6144 (6 GB), 7168 (7 GB), 8192 (8 GB)
 - 2048 (2 vCPU) - Available memory values: Between 4096 (4 GB) and 16384 (16 GB) in increments of 1024 (1 GB)
 - 4096 (4 vCPU) - Available memory values: Between 8192 (8 GB) and 30720 (30 GB) in increments of 1024 (1 GB)
- **ContainerMemory** – RAM that will be allocated for a single container. This parameter depends on the value specified in the ContainerCpu field. Please be attentive.
- **ContainerPort** – container port that will be opened for the DataSunrise WebUI. You will be able to add more ports later.
- **ContainerProxyPort** – container port that will be opened if proxy mode is used.
- **ContainersCount** – specify how many containers you want to be launched.
- **Image** – repository from where you will be pulling the Docker Image.
- **Subnets** – define subnets where tasks will be started.
- **VPC** – define VPC where tasks will be started.
- **VpcCidr** – define VPC CIDR since it will be used as the allowed inbound rules for LB health checks to be passed.

3.3 Finishing the Deployment

Once all the sections are filled out, click **Next**. The following page will appear:

Configure stack options

Tags

You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack. [Learn more](#)

Permissions

Choose an IAM role to explicitly define how CloudFormation can create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses permissions based on your user credentials. [Learn more](#)

IAM role - optional
Choose the IAM role for CloudFormation to use for all operations performed on the stack.

Advanced options

You can set additional options for your stack, like notification options and a stack policy. [Learn more](#)

► **Stack policy**
Defines the resources that you want to protect from unintentional updates during a stack update.

► **Rollback configuration**
Specify alarms for CloudFormation to monitor when creating and updating the stack. If the operation breaches an alarm threshold, CloudFormation rolls it back. [Learn more](#)

► **Notification options**

► **Stack creation options**

Specify a Role that will be used for the Cloud Formation deployment. Just choose the role from the drop-down menu in the IAM Role field.

Check once again the parameters you have specified in the stack's settings. If everything is ok, just check the check box below and click Create Stack:

► **Quick-create link**

Capabilities

i The following resource(s) require capabilities: [AWS::IAM::ManagedPolicy, AWS::IAM::Role]

This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have custom names. Check that the custom names are unique within your AWS account. [Learn more](#)

☒ I acknowledge that AWS CloudFormation might create IAM resources with custom names.

This is it. If everything is done according to this guide, the stack creation will start and you will be redirected to the Events Page:

Here you can observe resource creation processes. As soon as the Cloud Formation stack receives the “CREATE_COMPLETE” status, you can proceed to the Outputs tab and see the information that can be used to connect to the DataSunrise Web Console or Target database endpoint.

4. Fault-tolerance and Recovery

Detecting an application fault

Application fault detection is one of the top-priority aspects of the infrastructure. The CloudFormation script has a built-in mechanism for detecting application faults. To activate this mechanism, you must specify their email address in the *AlarmEmail field (optional)* of the CloudFormation script’s settings and accept the AWS SNS newsletter as soon as the topic message is received during the deployment of the stack. AWS SNS topic will be triggered if DataSunrise service crashes or any downtime is detected.

In order to improve the fault detection efficiency, we recommend you to configure and use DataSunrise’s health check periodic task. This task checks a connection between DataSunrise and the target database, proxies and load balancer. To get notifications, you will need to configure an SMTP server and add a subscriber that will receive notifications about any faults detected. For detailed information, please refer to the DataSunrise User Guide sections “Health Check” and “Subscriber Settings”.

Conducting the recovery testing

The CloudFormation template is dedicated to deploying a fault-tolerant and recoverable stack. Fault-tolerance and recovery features are based on AWS Auto Scaling service policies, which contain predefined values for successfully scaling DataSunrise tasks using the HA configuration.

To test fault tolerance and recovery capabilities of a DataSunrise stack, you can:

1. Simulate CPU usage by running traffic to the proxy server and checking the scaling function.
2. Manually terminate your task instance in order to check if a new task starts back with the DataSunrise service already configured to act as a proxy for a target database.

5. IAM Role and Policies

Establishing security settings properly is an obligatory requirement for any software and cloud applications are not exception. Setting up policies and security groups of the cloud infrastructure components can be a challenging and time-consuming task.

DataSunrise CloudFormation script includes all the settings needed for deployment of a production-ready DataSunrise Security Suite in no time and without any manual actions.

The security policies created by the CloudFormation script are based on the least privilege principle and do not provide excessive access to the resources of the environment where DataSunrise is deployed.

In order to use CloudWatch alarms and store backups and logs of DataSunrise, the policies are attached to the **TaskRole** IAM Role, which is attached to the tasks of DataSunrise. The creation of the role is also automated via the CloudFormation script and does not require any manual actions to be performed.

FAQ

Q: I deployed the CloudFormation script and when I am trying to connect to the Web Console using the Load Balancer endpoint and port 11000, I am having a loop. What's the problem?

A: The situation you've encountered is connected with the Load Balancer round-robin system and self-signed certificates that are used for DataSunrise Web Console. As soon as you accept both of the Certificates the issue should be solved. Sometimes loops happen and it's normal behavior.

However, we suggest using Safari in case you're running a macOS, or Google Chrome browser in case you are running Linux or Windows OS.

Q: How to perform Key-Rotation to keep my connection secure?

A: DataSunrise's Web Console features a section to keep your private keys rotated and up-to-date. In order to rotate your keys, simply change them in the WebUI -> Configuration -> SSL Key Groups. For more information please refer to the DataSunrise User Guide section "SSL Key Groups".

Q: I got the following errors in System Events: "DS_31001E: AWS Metering Service failure!" I am also unable to create and use any Rule and get the following error: "New rules are no longer covered by license". These two errors are displayed simultaneously.

A: When using Hourly Billing license type, it is necessary to have public access your AWS Marketplace Metering endpoint. This can be done by attaching Internet Gateway, NAT Gateway to your private subnet route table or using HTTP Proxy with Internet Access. In case none of these options are permitted, please use the BYOL License type. To get a BYOL license, please contact DataSunrise support team.