

# CloudFormation Installer Kit

28/January/2021



This is sample code and does not form part of normal DataRobot distribution and its associated support process.  
**Use with Caution**

A review of the SecurityGroup-Stacks are required before execution.



These CloudFormation Templates do not support CI or CD

# CloudFormation Installer Kit

DataRobot 6.3.2

---

<b>Summary</b>	<b>4</b>
<b>Requirements</b>	<b>5</b>
DataRobot Installation Email	5
AWS Requirements	5
Command Line Requirements	5
<b>Environment Details</b>	<b>6</b>
<b>Cluster Actions</b>	<b>7</b>
<b>Available Clusters</b>	<b>7</b>
<b>Available DataRobot Features</b>	<b>8</b>
<b>Advanced CLI Usage</b>	<b>10</b>
<b>Getting Started With the AWS Console</b>	<b>14</b>
<b>Post-Flight</b>	<b>21</b>
<b>Sample Files</b>	<b>21</b>
<b>Enable EBS Snapshots</b>	<b>22</b>
<b>Enable Fast Snapshot Restore</b>	<b>25</b>
<b>Enable AutoScaling</b>	<b>25</b>
<b>Appendix: Required IAM Roles</b>	<b>29</b>
<b>Appendix: cleanup-ssm-parameters.sh</b>	<b>29</b>
<b>Appendix: cleanup-autoscaling.sh</b>	<b>30</b>

---

## Summary



This kit leads the user through the automated process of using the Amazon Web Services CloudFormation services to create a working DataRobot cluster using the following high level process:

- Confirm user has all required permissions
- Create a private bucket
- Download the DataRobot CloudFormation kit to a safe location and unpack it
- Upload all files to the new private bucket
- By default, DataRobot is configured to use S3Bucket storage and the storage directory will be created at build time. Please note, MinIO and Gluster are made available if required
- Execute the stack, which creates:
  - A Security Group, with the appropriate DataRobot ports opened
  - An IAM Role, with the required permissions
  - If used, the ELK node
  - All EC2 instances required for the cluster
  - A fully functioning and ready DataRobot cluster
- The cluster is ready when the admin username and password are written to /home/datarobot/admin\_password
- After initial login, the /home/datarobot/admin\_password is safe to delete.
- Look for a working <http://<app server>/availabilitymonitor/v1/health>
- These templates currently use the Small Maps set for the Location AI
- Allows for S3:// url file ingestion

# Requirements



**Please note:** Be sure to include your AWS administration team when preparing these templates to review the IAM Roles and SecurityGroup settings

DataRobot Installation Email	The email sent from <a href="mailto:support@datarobot.com">support@datarobot.com</a> , containing the time expiring links to the DataRobot software repository.						
AWS Requirements	<table> <tr> <td><b>Required Infrastructure</b></td><td> <ul style="list-style-type: none"> <li>• An existing VPC</li> <li>• 1 subnet for SingleNode/PoC/PreProd</li> <li>• 3 subnets for Enterprise</li> </ul> </td></tr> <tr> <td><b>General Requirements</b></td><td>S3 bucket for template and data storage</td></tr> <tr> <td> <b>IAM Roles</b>  Refer to the <a href="#">appendix</a> for the complete list </td><td> <b>Global:</b> cloudformation, ec2, lambda, ssm &amp; s3  <b>AutoScaling:</b> autoscaling, cloudwatch  <b>Scheduler:</b> logs &amp; events </td></tr> </table> <p><b>An ip allowed access to the cluster.</b>  Currently using the result of:</p> <pre>echo \$(curl --silent http://whatismyip.akamai.com)/32</pre>	<b>Required Infrastructure</b>	<ul style="list-style-type: none"> <li>• An existing VPC</li> <li>• 1 subnet for SingleNode/PoC/PreProd</li> <li>• 3 subnets for Enterprise</li> </ul>	<b>General Requirements</b>	S3 bucket for template and data storage	<b>IAM Roles</b> Refer to the <a href="#">appendix</a> for the complete list	<b>Global:</b> cloudformation, ec2, lambda, ssm & s3 <b>AutoScaling:</b> autoscaling, cloudwatch <b>Scheduler:</b> logs & events
<b>Required Infrastructure</b>	<ul style="list-style-type: none"> <li>• An existing VPC</li> <li>• 1 subnet for SingleNode/PoC/PreProd</li> <li>• 3 subnets for Enterprise</li> </ul>						
<b>General Requirements</b>	S3 bucket for template and data storage						
<b>IAM Roles</b> Refer to the <a href="#">appendix</a> for the complete list	<b>Global:</b> cloudformation, ec2, lambda, ssm & s3 <b>AutoScaling:</b> autoscaling, cloudwatch <b>Scheduler:</b> logs & events						
For AWS Admins or Power Users							
Command Line Requirements	<ul style="list-style-type: none"> <li>• <a href="#">The AWS CLI</a></li> <li>• AWS Access and Private Keys</li> <li>• See '<a href="#">Advanced CLI Usage</a>' for more</li> </ul>						

## Environment Details

All environments are provisioned as follows:

- Using RHEL/CentOS 7.6+ & <8
- A DataRobot volume mounted at /opt/datarobot using ext4 and able to survive a reboot.
- A symlink is created from /var/lib/docker to /opt/datarobot/docker
- Secrets are enabled
- IPv6 is disabled
- The ability to add a KMS key to the DataRobot volume
- Generate the minimum viable IAM permissions to generate the cluster
- Blacklist everything except ports listed in the Install Guide based on storage type
- Tag each instance with:
  - An appropriate node name, i.e.: AppNode or DataNode1, etc.
  - The name of the stack that built it
  - The owner, group or org that using this instance of DataRobot  
Defaults to the portable version of ``whoami``
  - The username that actually launched the command line build process
  - Any custom tags given by the builder at the command line or CF user interface.
- Provide the private IPs and instance IDs of each node in the cluster
- Provide the ssh connection string to the App Node
 

```
Ex: ssh -i <ssh-key>.pem <username>@<App node>
```
- Provide the URL to connect via the browser
 

```
http://<App node>
```
- Provide the availability monitor url
 

```
curl -v http://<App node>/availabilitymonitor/v1/health
```
- The required settings for the Scheduler or AutoScaling
- Display the public IP's of the App node and/or the DPE
- By default, the templates will use S3 Bucket storage, but MinIO and Gluster are available
- By default, the cluster will use an S3 Bucket directory to store the system back up on all the non-modeling nodes everyday at midnight. The other options are EBS Snapshots or none.
- Upon a successful installation, the secrets information will be saved to the bucket.

## Cluster Actions

The user has the option of either creating a new cluster, restoring a cluster from a set of S3 Bucket locations or EBS Snapshots. Coming soon, upgrading an existing DataRobot cluster.

Where applicable, this will be denoted with the “<Action>” tag.

## Available Clusters

Listed below are the available environments that can be created by this stack.

When using debug mode, all nodes = m5.2xlarge

<b>SingleNode</b>	<b>For debugging/development use only.</b> <b>Not to be used by more than 1 user, does not provide the AI Catalog, a DPE and no MM.</b> 1 r5d.12xlarge node using a 2 TB hd
<b>PoC</b>	<b>Small Scale DataRobot functionality testing, by &lt;= 3 concurrent users.</b> 1 AppDataNode: r5.2xlarge with 2TB 1 ElasticSearch Node: m5.xlarge with 1TB 2 ModelingNodes: r5.4xlarge with 100GB 1 ModMan: m5.xlarge with 1TB 1 DPE: r5.xlarge with 100GB
<b>PreProd</b>	<b>Small environment with App and Data node split.</b> <b>Best used for small sets of users, no more than 3 concurrently</b> 1 AppNode: r5.xlarge with 250GB 1 DataNode: c5.xlarge with 2TB 1 ElasticSearch Node: m5.xlarge with 1TB 2 ModelingNodes: r5.4xlarge with 100GB 1 ModelingOnlyNodes: r5.4xlarge with 100GB 1 ModMan: m5.xlarge with 1TB 1 DPE: r5.xlarge with 100GB
<b>Enterprise</b> A Special License is Required	<b>This is the recommended configuration for an Enterprise Class Production Environment</b> 1 App Node: r5.x2large with 250GB 3 Data Nodes: c5.xlarge with 2TB 3 ElasticSearch Nodes: m5.xlarge with 1TB 2 ModelingDSSNodes: r5.4xlarge with 100GB 1 ModelingOnlyNodes: r5.4xlarge with 100GB 3 ModMan: m5.xlarge with 1TB 1 or 3 DPE: r5.xlarge with 100GB based on UseDPELoadBalancer (Route53 and config.yaml update required)

## Available DataRobot Features

Out of the box, the PreProd and Enterprise clusters are configured for AutoScaling and all cluster types are ready for the EBS Scheduler. Below, please find the defaults

### AutoScaling

Auto Scaling Group			
These settings allow for a maximum of 52 total workers.			
<b>Name:</b> <Stack Name>-AutoScaling-Group	<b>Desired Capacity:</b> 1	<b>Min:</b> 1	<b>Max:</b> 20

Scale Down Policy	
<b>Name</b>	<Stack Name>--ScaleDown-Policy
<b>Execute Policy when</b>	<Stack Name>-ScaleDown-Modeling-Capacity-Alarm breaches the alarm threshold: <watchedmetric> <= <thresholdremove> for <evaluationperiodsremove> consecutive periods of <periodremove> seconds for the metric dimensions
<b>Take the action</b>	Remove <scalingadjustmentremove> capacity units

Scale Up Policy	
<b>Name</b>	<Stack Name>--ScaleUp-Policy
<b>Execute Policy when</b>	<Stack Name>-ScaleUp-Modeling-Capacity-Alarm breaches the alarm threshold: <watchedmetric> > <thresholdadd> for <evaluationperiodsadd> consecutive periods of <periodadd> seconds for the metric dimensions
<b>Take the action</b>	Remove <scalingadjustmentremove> capacity units

Alarm Defaults			
<b>evaluationperiodsadd:</b> 2	<b>periodadd:</b> 30	<b>scalingadjustmentadd:</b> 1	<b>thresholdadd:</b> 0.75
<b>evaluationperiodsremove:</b> 20	<b>periodremove:</b> 240	<b>scalingadjustmentremove:</b> -1	<b>thresholdremove:</b> 0.25

# Backups

## S3 Bucket cron job

This backup style will use a sub directory in the existing S3 Bucket to store all the data from the non-modeling nodes, by shutting down the service, running the aws s3 sync command and turning the service back on every Monday through Friday at 12am on the local node.

This option is recommended if cost is an issue.

## EBS SnapShot Scheduler

This function will find the tagged instances and create a snapshot of them and delete any snapshots older than 5 days.

This option is recommended when security is paramount.

### Default Configuration

<b>ScheduleCustomTagName</b>	scheduler:ebs-snapshot
<b>ScheduleExpression</b>	cron(0 0 ? * MON-FRI *) # Midnight UDT Monday - Friday

### Node Types to SnapShot per Cluster

	AppData	App	App1	Data	Data1	Data2	Data3	MM	MM1	MM2	MM3	ES	ES1	ES2	ES3
<b>Single Node</b>	X														
<b>PoC</b>	X							X				X			
<b>PreProd</b>		X		X				X				X			
<b>Enterprise</b>			X		X	X	X		X	X	X		X	X	X



## Advanced CLI Usage

This is the recommended method to interact with the AWS CloudFormation system as it provides the quickest and simplest method of creating the cluster.



**Please note:** All of the following commands assume the user has proper AWS and SSH Keys to access the given environment. Otherwise, Contact your AWS IT Group.

### Log onto the AWS command line

The simple way to log onto AWS is to do as follows

```
export AWS_ACCESS_KEY_ID=<AWS_ACCESS_KEY_ID>
export AWS_SECRET_ACCESS_KEY=<AWS_SECRET_ACCESS_KEY>
```

**Where:**

<AWS\_ACCESS\_KEY\_ID> = The AWS Access key provided by your organization.

<AWS\_SECRET\_ACCESS\_KEY> = The AWS Secret key provided by your organization.



**Please note:** If your organization has a different login process, please adjust as required.

### Stage the Files

After you have saved the kit, use the process below to get these files ready.

```
mkdir -p /my/safe/directory
cp /path/to/DataRobot-CloudFormation-<Env>-Kit.<timestamp>.tar.gz /my/safe/directory
cd /my/safe/directory
tar xzvf DataRobot-CloudFormation-PoC-Kit.20200822.115659.tar.gz
```

### Confirm-AWS-Settings.py

This command walks the user through the process of finding the various bits of information in AWS required to execute DataRobot\_CloudFormation\_Kit.py

## Usage

```
./Confirm-AWS-Settings.py --help
usage: Confirm-AWS-Settings.py [-h] [--checks3] [--url URL]
```

Does AWS testing to ensure the user can execute the CloudFormation stack.  
Warning, can cause clutter if you do not have S3 bucket delete permissions.

optional arguments:

```
-h, --help  show this help message and exit
--checks3   Confirm S3 Bucket creation, modification and delete
--url URL   DataRobot download url provided by support@datarobot.com
```

DataRobot Copyright 2020

## Example

```
./Confirm-AWS-Settings.py --url "https://<DataRobot Download URL>"
=== Started Confirm-AWS-Settings.py ===
[?] Cluster action:: create
> create
  restore
  upgrade

[?] Please define your type of project?: PoC
  SingleNode
> PoC
  PreProd
  Enterprise

[?] Please select your region: us-east-1
  eu-west-2
  eu-west-1
  ap-northeast-2
  ap-northeast-1
  sa-east-1
  ca-central-1
  ap-southeast-1
  ap-southeast-2
  eu-central-1
> us-east-1
  us-east-2
  us-west-1
  us-west-2
```

```
[?] Select SSH Key: ssh-key-1
> ssh-key-1
ssh-key-2
ssh-key-3

[?] Select VPC: vpc-a1b2c3d4e5
vpc-1234567890
vpc-abcdefghij
> vpc-a1b2c3d4e5

[?] Select a subnet: subnet-3
subnet-1
subnet-2
> subnet-3

[?] Generated Cidr Block: 10.X.Y.0/22
[?] Configure for access from:: A.B.C.D/32
[?] Enable secret enforcement? (Y/n):

[?] Use KMS key on the DataRobot volume? (y/N): y

[?] Choose KMS Key: kms-key-4
kms-key-1
kms-key-2
kms-key-3
> kms-key-4

[?] Who is the owner of this cluster?: <current username>
[?] Enable the EBS Snapshot Scheduler for this cluster? (y/N):
[?] Enable Auto Scaling for this cluster? (y/N):
[?] Please provide the DataRobot download url: https://<DataRobot Download URL>
=====
Be sure to update the --owner and --url parameters as required
=====
./DataRobot_CloudFormation_Kit.py create --environment PoC --region us-east-1
--sshkey ssh-key-1 --vpc vpc-a1b2c3d4e5 --subnet subnet-3 --cidr "10.X.Y.0/22"
--externalip "A.B.C.D/32" --secretsenforced --encrypted --encryptionkey
"kms-key-4" --owner <current username> --url "https://<DataRobot Download URL>"
=====
Buildout completed in 0:00:59.502591
```



**Please note:** "<current username>" is the username of the user running the script. Be sure to update with the name of the sponsor of the DataRobot cluster

## DataRobot\_CloudFormation\_Kit.py

To use this package, you must be on a CentOS, RedHat or Mac system.  
Windows should work with Cygwin or the Linux sub-system  
you must use current, valid AWS ACCESS and SECRET key tokens.

After unpacking the DataRobot-CloudFormation-Kit.tar.gz to a safe location,  
Execute the script and follow the prompts

```
/my/safe/directory/Confirm-AWS-Settings.py --url "<DataRobot Download URL>"
```

If you ran the Confirm-AWS-Settings.py script, you can copy/paste the given command

## Usage

Use the following commands to find the latest usage for the given action

```
./DataRobot_CloudFormation_Kit.py --help
```

### Action Usage

```
./DataRobot_CloudFormation_Kit.py create --help
```

```
./DataRobot_CloudFormation_Kit.py describe --help
```

```
./DataRobot_CloudFormation_Kit.py restore --help
```

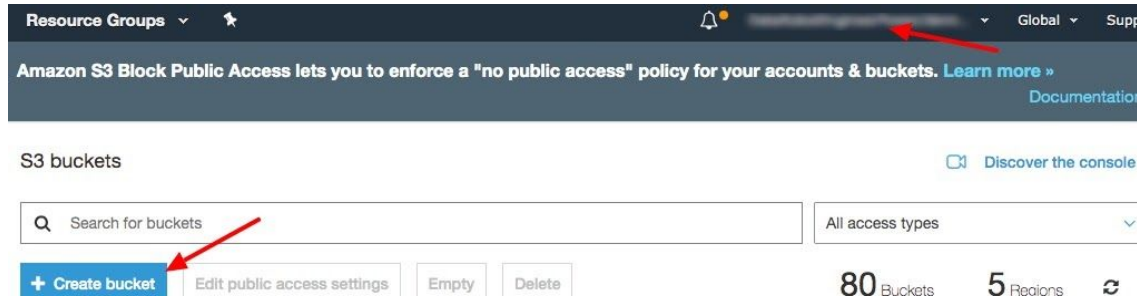
```
./DataRobot_CloudFormation_Kit.py delete --help
```

Once complete, please refer to the "[Post-Flight](#)" tasks for more

# Getting Started With the AWS Console

## Make the Bucket

1. Log into the AWS S3 console <https://s3.console.aws.amazon.com/s3/home>, confirm your user name, then click the "Create bucket" button.

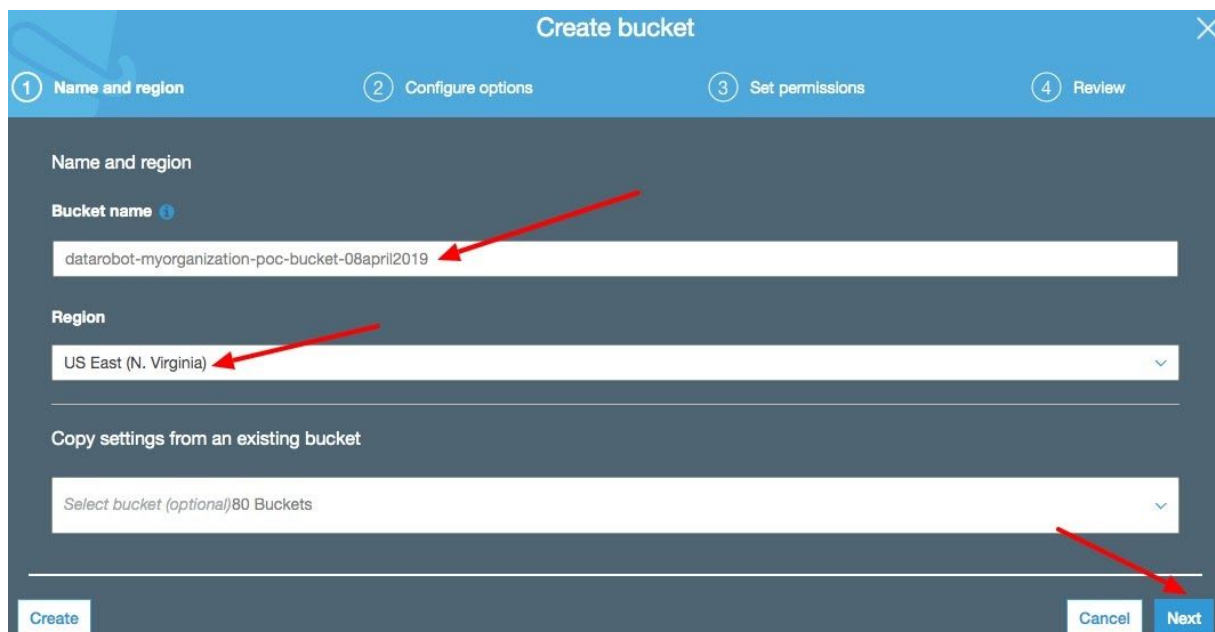


2. Name your bucket, select your region and click Next.

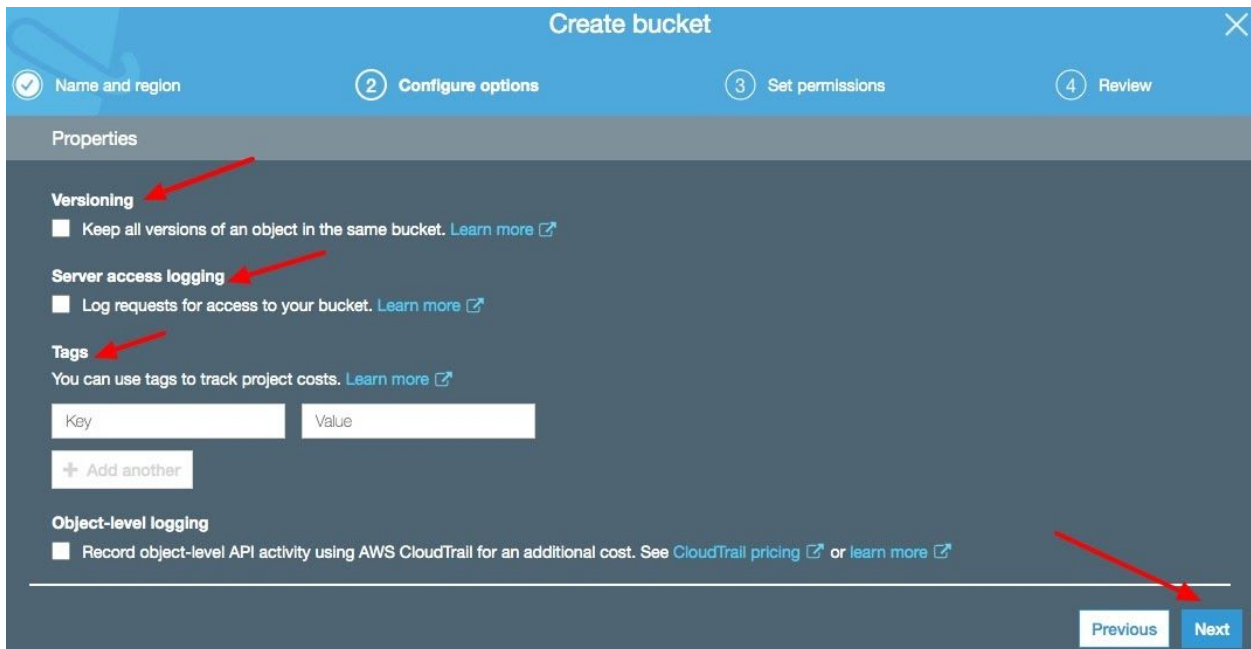
We recommend a variation on "datarobot-[myorganization]-[environment]-bucket-[date]", where:  
[myorganization] = Name of the sponsoring organization of this DataRobot installation.

[environment] = 1 of ["SingleNode", "PoC", "PreProd", "Enterprise"]

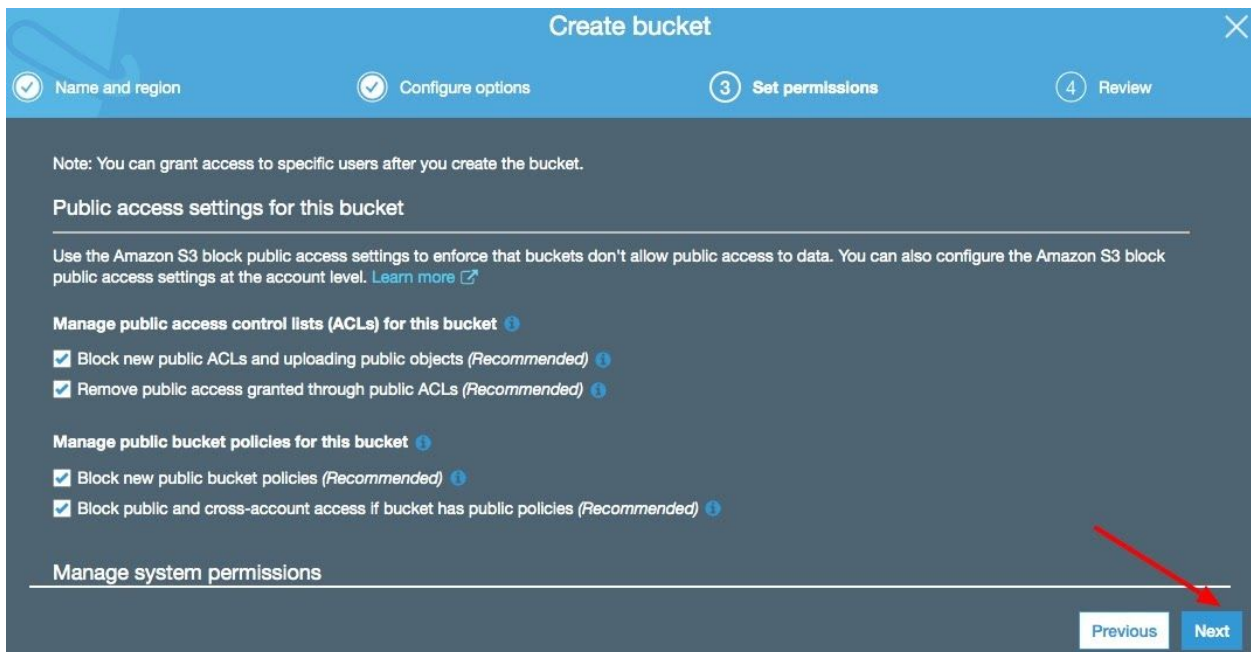
[date] = today's date in the format DDMonthnameYear to allow for versioning



3. You have the option to turn on versioning, logging or adding tags and click Next.



4. Unless your AWS group says otherwise, leave this alone and click Next



5 Review and click next.

## Fill the bucket

The next part of this process is to place the files contained in the DataRobot CloudFormation Kit archive into your new bucket.

### 1. Create a safe place to stage our files and enter it

```
mkdir -p /my/safe/directory
cd /my/safe/directory
```

### 2. Unpack the DataRobot tarball into this directory.

```
tar xzvf DataRobot-CloudFormation-Kit.20200822.115659.tar.gz
x Confirm-AWS-Settings.py
x DataRobot_CloudFormation_Kit.py
x EC2-Create-Child-Stack.yaml
x EC2-Create-Parent-Stack.yaml
x EC2-Restore-Child-Stack.yaml
x EC2-Restore-Parent-Stack.yaml
x ELK-Stack.yaml
x IAM-Stack.yaml
x Master-Create-Stack.yaml
x Master-Restore-Stack.yaml
x SecurityGroup-gluster-Stack.yaml
x SecurityGroup-minio-Stack.yaml
x SecurityGroup-s3bucket-Stack.yaml
x bin/
x bin/flush-old-docker.sh
x bin/reconf-repl.tpl
x etc/
x etc/X.Y.Z-Enterprise-config.tpl
x etc/X.Y.Z-PoC-config.tpl
x etc/X.Y.Z-PreProd-config.tpl
x etc/X.Y.Z-SingleNode-config.tpl
x etc/README.md
x readme.txt
x test-stack.yaml
```

### 3. Find your new bucket on <https://s3.console.aws.amazon.com/s3/home>

S3 buckets [Discover the console](#)

All access types ▾

[+ Create bucket](#)
[Edit public access settings](#)
[Empty](#)
[Delete](#)

**1** Buckets    **1** Regions

<input type="checkbox"/>	Bucket name ▾	Access ⓘ ▾	Region ▾	Date created ▾
<input type="checkbox"/>	datarobot-...-poc-bucket-08april2019	Bucket and objects not public	US East (N. Virginia)	Apr 8, 2019 1:52:56 PM GMT-0400

5. Upload the file set. This can be done by dragging and dropping all the files into the bucket or click the upload button and select your files.

6. Save the Object URL of the Master-<Action>-Stack.yaml file by clicking on it and saving the Object URL to the clipboard.

## Master-Create-Stack.yaml Latest version ▾

Overview
Properties
Permissions
Select from

[Open](#)
[Download](#)
[Download as](#)
[Make public](#)
[Copy path](#)

**Owner**  
awssupport

**Last modified**  
Jan 2, 2020 1:45:38 PM GMT-0500

**Etag**  
56c4a65983d6c9b1db41cd08bbecabd5

**Storage class**  
Standard

**Server-side encryption**  
None

**Size**  
41.2 KB

**Key**  
Master-Create-Stack.yaml

**Object URL**  
https://s3.amazonaws.com/datarobot-...-preprod-5.3.0-cloudformation/Master-Create-Stack.yaml

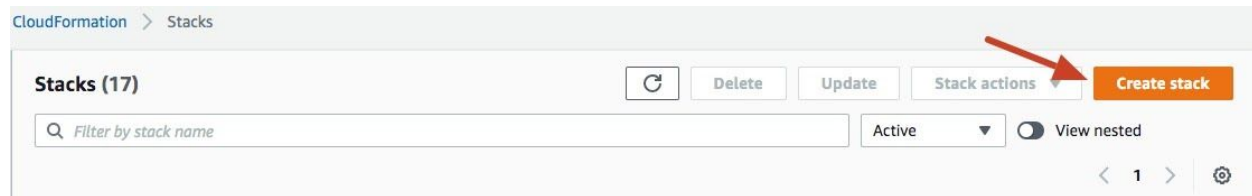


## Use the Bucket

Now that we have our bucket and the DataRobot CloudFormation file set in place in the bucket, we get to use them via CloudFormation

1. Go to the CloudFormation console, via:

<https://console.aws.amazon.com/cloudformation/home> and click on the “Create stack” button



2. Click the “Amazon S3 URL” field and psat the path you saved earlier and click Next

### Create stack

#### Prerequisite - Prepare template

Prepare template  
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☒ Template is ready
 ☐ Use a sample template
 ☐ Create template in Designer

#### Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source  
Selecting a template generates an Amazon S3 URL where it will be stored.

☒ Amazon S3 URL
 ☐ Upload a template file

Amazon S3 URL **Paste URL here**

*https://*

Amazon S3 template URL

S3 URL: Will be generated when URL is provided

3. Fill out the form and click Next.

- The **Create Stack** will ask the user to provide the Download URL

**DownloadURL**

Link to the time expiring DataRobot installation archive

- The **Restore Stack** will ask for SnapShot IDs for the given node. For example:

**Volume Snapshot settings**

**App1SnapShot**

EBS SnapShot ID for the App Node

**DataNode1SnapShot**

EBS SnapShot ID for DataNode1

- The **Update Stack** requires both
- **All Templates** require:

**VpcId**

Virtual Private Cloud Id

**Subnet**

Subnet ID for the first App node and DPE

### SSHKey

Valid AWS Key for SSH connectivity

### CidrBlock

The IPv4 CIDR that can communicate with these EC2 instances. Please refer to

[https://console.aws.amazon.com/vpc/home?](https://console.aws.amazon.com/vpc/home?subnets&sort=SubnetId&sort=SubnetId&subnets:sort=SubnetId&region=us-east-1#subnets:sort=SubnetId)

[subnets&sort=SubnetId&sort=SubnetId&subnets:sort=SubnetId&region=us-east-1#subnets:sort=SubnetId](https://console.aws.amazon.com/vpc/home?subnets&sort=SubnetId&sort=SubnetId&subnets:sort=SubnetId&region=us-east-1#subnets:sort=SubnetId) for more

### ExternalIPAddress

IP Address of external connection, with the "/32" post fix. <https://www.whatsmyip.org/>

4. Click the Next button unless you have a directive to add an IAM Role or Tags.

5. Review your settings, check the boxes and click Create

#### Capabilities

**The following resource(s) require capabilities: [AWS::CloudFormation::Stack]**

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.](#)

For this template, AWS CloudFormation might require an unrecognized capability: CAPABILITY\_AUTO\_EXPAND. Check the capabilities of these resources.

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

☒ I acknowledge that AWS CloudFormation might require the following capability:  
CAPABILITY\_AUTO\_EXPAND

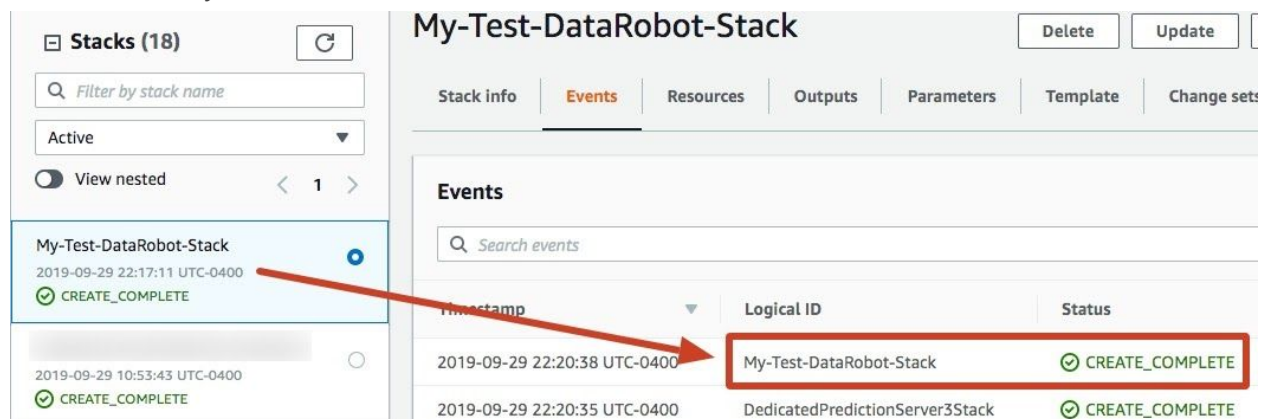
Cancel

Previous

Create change set

Create stack

If all went well, you should see:



**Stacks (18)**

Filter by stack name

Active

View nested

My-Test-DataRobot-Stack

2019-09-29 22:17:11 UTC-0400

CREATE\_COMPLETE

2019-09-29 10:53:43 UTC-0400

CREATE\_COMPLETE

**My-Test-DataRobot-Stack**

Delete Update

Stack info Events Resources Outputs Parameters Template Change sets

**Events**

Search events

Timestamp	Logical ID	Status
2019-09-29 22:20:38 UTC-0400	My-Test-DataRobot-Stack	CREATE_COMPLETE
2019-09-29 22:20:35 UTC-0400	DedicatedPredictionServer3Stack	CREATE_COMPLETE

## Post-Flight

At this point, your DataRobot cluster is being created by the various components of AWS. To see the elements available to you, click the Outputs tab.

Use the SSHConnectionString to log onto the App Node for further maintenance.

```
ssh -i /path/to/my/SSHKey.pem ec2-user@11.22.33.44
```

Review the installation process using the /home/datarobot/install.log

```
sudo tail -f /home/datarobot/install.log
```

Once completed, the admin username and password will be stored as JSON in /tmp/admin\_password and once copied can be deleted off the node as it is no longer required.

```
sudo cat /home/datarobot/admin_password  
{ "user": "localadmin@datarobot.com", "password": "<24 char alphanumeric string>" }
```

## Sample Files

URL	Target Variable
<a href="https://s3.amazonaws.com/datarobot_public_datasets/10k_diabetes.xlsx">https://s3.amazonaws.com/datarobot_public_datasets/10k_diabetes.xlsx</a>	readmitted
<a href="https://s3.amazonaws.com/datarobot_test/kickcars-sample-200.csv">https://s3.amazonaws.com/datarobot_test/kickcars-sample-200.csv</a>	isBadBuy

# Enable EBS Snapshots



## WARNING!

You must enable the “Fast Snapshot Restore” on the snapshots to be used! Otherwise, the system will fail! See below for more.

<https://aws.amazon.com/blogs/aws/new-amazon-ebs-fast-snapshot-restore-fsr/>



## WARNING #2!

Before using this feature in Enterprise mode, you must contact AWS support and request an increase to the EBS Snapshot count to 20 to allow for what your cluster needs

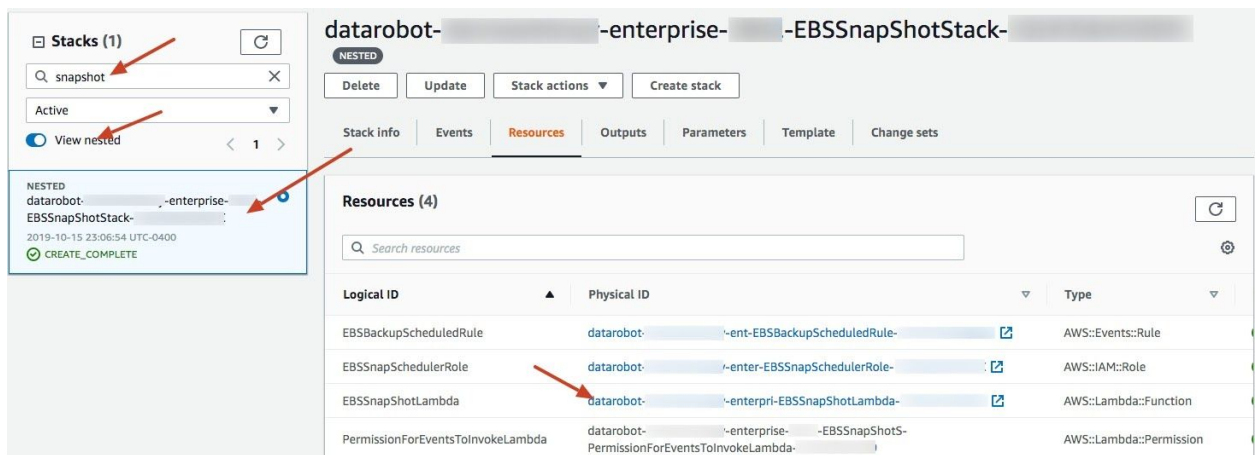
<https://aws.amazon.com/premiumsupport/knowledge-center/ebs-snapshot-billing/>

To complete the process of enabling the AWS EBS Snapshot functionality, we need to finish the configuration of the Lambda function and the CloudWatch Rule. To accomplish this task, use the following process.

- 1) On the Outputs tab of the CloudFormation template, copy the value of the **LambdaConfiguration** key, as shown.

LambdaConfiguration	<code>{ "RequestType": "Create", "StackName": "datarobot-~-preprod" }</code>	Configuration for the Lambda Functions
---------------------	--	--

- 2) In the Stacks list column, click the **View nested** slider and enter “snapshot” as the search term. Select the correct result (it should have a similar name as the main stack), then click the Physical ID link for the EBSSnapShotLambda Logical ID.



The screenshot shows the AWS CloudFormation console. On the left, the 'Stacks (1)' list is visible with a search bar containing 'snapshot'. The 'View nested' slider is active. A result for 'datarobot-~-enterprise-~-EBSSnapShotStack-~' is selected. On the right, the 'Resources (4)' tab is open for this stack. The resources listed are:

Logical ID	Physical ID	Type
EBSSnapShotLambda	datarobot-~-enterprise-~-EBSSnapShotLambda-	AWS::Lambda::Function
EBSSnapShotSchedulerRole	datarobot-~-enter-EBSSnapShotSchedulerRole-	AWS::IAM::Role
EBSSnapShotSchedulerRule	datarobot-~-ent-EBSSnapShotSchedulerRule-	AWS::Events::Rule
PermissionForEventsToInvokeLambda	datarobot-~-enterprise-~-EBSSnapShotS-PermissionForEventsToInvokeLambda-	AWS::Lambda::Permission

- 3) Click the “Configure test events” drop down box, enter **Create** in the **Event name** field and paste the json saved in step 1 into the textarea as shown and click the **Create** button

Configure test event ×

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

☒ Create new test event  
☐ Edit saved test events

Event template

Hello World

Event name

Create

```

1 {
2   "RequestType": "Create",
3   "StackName": "datarobot-~-enterprise-"
4 }
        
```

- 4) Go to <https://console.aws.amazon.com/cloudwatch/>, click **Events > Rules**, search for **EBSBackupScheduledRule** and click the rule that was made by your stack.

CloudWatch  
Dashboards  
Alarms  
ALARM  
INSUFFICIENT  
OK  
Billing  
Events  
**Rules**  
Event Buses

### Rules

Rules route events from your AWS resources for processing by selected targets. You can create, edit, and delete rules.

Create rule Actions

Status All Name EBSBackupScheduledRule

Status	Name
<span>○</span>	datarobot-~-ent-EBSBackupScheduledRule-

If All went well, you should see something like

Schedule Cron expression 0 0 ? \* MON-FRI \*

Next 10 Trigger Date(s)

- 1. Mon, 06 Jan 2020 00:00:00 GMT
- 2. Tue, 07 Jan 2020 00:00:00 GMT
- 3. Wed, 08 Jan 2020 00:00:00 GMT
- 4. Thu, 09 Jan 2020 00:00:00 GMT
- 5. Fri, 10 Jan 2020 00:00:00 GMT
- 6. Mon, 13 Jan 2020 00:00:00 GMT
- 7. Tue, 14 Jan 2020 00:00:00 GMT
- 8. Wed, 15 Jan 2020 00:00:00 GMT
- 9. Thu, 16 Jan 2020 00:00:00 GMT
- 10. Fri, 17 Jan 2020 00:00:00 GMT

Status Enabled

Description Rule to trigger the EBS SnapShot Backup lambda function on a schedule

Monitoring [Show metrics for the rule](#)

## How To OverRide

Via the Lambda Function configuration, click the “Configure test events” drop down box, enter **Create** in the **Event name** field and paste the json saved in step 1 into the textarea as shown and click the **Create** button

```
{
  'RequestType': 'create',
  'AutoDelete': 'true',
  'RetentionPeriod': 7,
  'CustomTagName': 'scheduler:ebs-snapshot',
  'RegionName': 'us-east-1'
}
```

Or Via the Rules UI

Go to <https://console.aws.amazon.com/cloudwatch/>, click **Events** > **Rules**, search for **EBSBackupScheduledRule** and click the rule that was made by your stack, click the **Actions** button, then **edit** and modify the **Event Source** as required.

### Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

☐ Event Pattern ⓘ ☒ Schedule ⓘ

☐ Fixed rate of

☒ Cron expression

[Learn more about CloudWatch Events schedules.](#)

► Show sample event(s)



## Enable Fast Snapshot Restore



### WARNING!

Do not enable this for all the snapshots in your environment as it is a metered, paid function where you must request an increase from AWS

To enable this

- 1) go to <https://console.aws.amazon.com/ec2/>, select **Elastic Block Store > Snapshots**
- 2) Select one of your snapshots. They should look something like:
  - AppNode-Data-Volume-SnapShot
  - DataNode-Data-Volume-SnapShot
  - ESNode-Data-Volume-SnapShot
  - ModelManagement-Data-Volume-SnapShot
- 3) Click the **actions** button and select the **Manage Fast Snapshot Restore** option from the dropdown
- 4) Select the appropriate availability zone. It should be the same as your subnet
- 5) Repeat steps 2-4 until all nodes selected

## Enable AutoScaling

- 1) On the Outputs tab of the CloudFormation template, copy the value of the **LambdaConfiguration** key, as shown.

LambdaConfiguration	<code>{ "RequestType": "Create", "StackName": "datarobot- -preprod" }</code>	Configuration for the Lambda Functions
---------------------	--	--

- 2) In the **Stacks** list column, click the **View nested** slider, enter the search term **"autoscaling"** and select your result (it should have a similar name as the main stack), Then click the Physical ID link for the **AutoScalingLambda** Logical ID.

The screenshot shows the AWS CloudFormation console. On the left, the 'Stacks' list is visible with a search filter 'autoscaling' and a 'View nested' slider. A red arrow points to the 'View nested' slider, and another red arrow points to the 'Nested' column. The main panel shows the details of the 'datarobot-  
-preprod-AutoScalingStack-' stack. The 'Resources' tab is selected, showing a table of resources. A red arrow points to the 'Physical ID' column for the 'AutoScalingLambda' resource.

Logical ID	Physical ID	Type	Status
AutoScalingLambda	datarobot-denniswhitney-preprod-AutoScalingLambda-16HT7FEEMXQC6	AWS::Lambda::Function	CREATE_COMPLETE
AutoScalingRole	datarobot-denniswhitney-preprod-AutoScalingRole-6N8VZOPJ97GX	AWS::IAM::Role	CREATE_COMPLETE



- Click the “Configure test events” drop down box, enter **Create** in the **Event name** field and paste the json saved in step 1 into the textarea as shown and click the **Create** button. Then click the **Test** button to execute the function

Configure test event ×

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

☒ Create new test event  
☐ Edit saved test events

Event template

Hello World ▼

Event name

Create

```

1 {
2   "RequestType": "Create",
3   "StackName": "datarobot-        -enterprise-"
4 }
```

If all went well, you should see the following:

- The Modeling Only Node being shut down
- The AMI being created
- The Launch Group generation
- The AutoScaling Group configuration
- Defining the ScaleUp and ScaleDown Cloudwatch Alarms and Policies
- The new AutoScaledWorker node appearing in the Instance list
- The Modeling Only Node being turned back on

If you see the following error:

✖ Execution result: failed ([logs](#))

▼ Details

The area below shows the result returned by your function execution. [Learn more](#) about returning results from your function.

```
{
  "errorMessage": "2020-01-03T18:25:57.228Z cf7bc18e-d008-4db3-b98a-711a9edd72c6 Task timed out after 900.10 seconds"
}
```

It typically means the AMI creation is taking an abnormally long period of time to generate.

datarobot- -preprod-AutoScaledWorkerImage

ami-

7/...

Private

pending

If you wait till the **status = available** then going back to the AutoScaling Lambda page, click the **Create** drop down, click the “**Configure test events**” option and update the event as shown, then click the **Save** button and then the **Test** button to execute the function

```
{
  "RequestType": "Create",
  "StackName": "datarobot-<owner name>-preprod",
  "SkipAMI": "true"
}
```

## How To OverRide

If it becomes necessary to execute this function manually, use the template below to see what is available. Please refer to [Scale Up Policy](#) or [Scale Down Policy](#) for the variable's details.

```
{
  "RequestType": "Create",
  "StackName": "datarobot-<owner>-<env>",
  "AutoScaleGroup": "datarobot-<owner>-<env>-AutoScaling-Group",
  "AutoScaleCap": 1,
  "AutoScaleMax": 20,
  "AutoScaleMin": 1,
  "ClusterType": "PreProd",
  "EvaluationPeriodsAdd": 2,
  "EvaluationPeriodsRemove": 20,
  "LaunchConfigName": "datarobot-<owner>-<env>-AutoScaling-LC",
  "MetricNamespace": "DataRobot/AutoScaling/<owner>Testing",
  "PeriodAdd": 30,
  "PeriodRemove": 240,
  "Region": "us-east-1",
  "Retries": 1200,
  "RetryDelay": 15,
  "ScaleDownAlarm": "datarobot-<owner>-<env>-ScaleDown-Modeling-Capacity-Alarm",
  "ScaleDownPolicy": "datarobot-<owner>-<env>-ScaleDown-Policy",
  "ScaleUpAlarm": "datarobot-<owner>-<env>-ScaleUp-Modeling-Capacity-Alarm",
  "ScaleUpPolicy": "datarobot-<owner>-<env>-ScaleUp-Policy",
  "ScalingAdjustmentAdd": 1,
  "ScalingAdjustmentRemove": -1,
  "Subnet": "<subnet-id>",
  "SkipAMI": "false",
  "ThresholdAdd": 0.75,
  "ThresholdRemove": 0.25,
  "WatchedMetric": "ClusterUtilization",
  "WorkerImageName": "datarobot-<owner>-<env>-AutoScaledWorkerImage",
  "WorkerName": "datarobot-<owner>-<env>-AutoScaledWorker",
  "WorkerType": "auto"
}
```

## Appendix: Required IAM Roles

Below is the list of required permissions for this template set to work correctly

cloudformation:DescribeStacks	elasticloadbalancing:*	s3:AbortMultipartUpload
ec2:CreateSnapshot	lambda:CreateFunction	s3:ListBucketVersions
ec2:CreateTags	lambda:DeleteFunction	s3:ListBucket
ec2:DeleteSnapshot	lambda:GetFunctionConfiguration	s3:DeleteObject
ec2:DescribeInstances	logs:CreateLogGroup	s3:GetBucketLocation
ec2:DescribeInstanceStatus	logs:CreateLogStream	s3:ReplicateDelete
ec2:DescribeRegions	logs:PutLogEvents	s3:ListMultipartUploadParts
ec2:DescribeSnapshots	s3:ListAllMyBuckets	ssm:DescribeParameters
ec2:DescribeTags	s3:Get*	ssm:PutParameter
ec2:DescribeVolumes	s3:PutObject	ssm:GetParameter
events:PutRule	s3:ListBucketMultipartUploads	ssm:DeleteParameter

## Appendix: cleanup-ssm-parameters.sh

If you find DataRobot AWS parameters that are unaccounted for, the script below will ask to delete each one for the given region and can be found in the tools directory

### Example Output

```
tools/cleanup-leftover-parameters.sh
Delete /datarobot-<somevalue>-preprod/datarobot-public-key [y/n]: y
"It's gone."
Delete /datarobot-<someothervaluevalue>-poc/datarobot-public-key [y/n]: Y
"It's gone."
Delete /datarobot-<athirdvalue>-poc-5855/datarobot-public-key [y/n]: n
Skipped
Delete /datarobot<afourthvalue>-poc/datarobot-public-key [y/n]: N
Skipped
```

## Appendix: cleanup-autoscaling.sh

Every so often, users forget to clean out the AutoScaling artifacts before deleting the DataRobot CloudFormation templates. The **tools/cleanup-autoscaling.sh** script will ask the user if they really want to delete the AutoScaling items and if yes, delete them.

### Example output

```
./cleanup-autoscaling.sh <env> <owner>
#=====
Launch Config      => datarobot-<owner>-<env>-AutoScaling-LC
AutoScaling Group => datarobot-<owner>-<env>-AutoScaling-Group
Scale Up Alarm     => datarobot-<owner>-<env>-ScaleUp-Modeling-Capacity-Alarm
Scale Up Policy    => datarobot-<owner>-<env>-ScaleUp-Policy
Scale Down Alarm   => datarobot-<owner>-<env>-ScaleDown-Modeling-Capacity-Alarm
Scale Down Policy  => datarobot-<owner>-<env>-ScaleDown-Policy
Worker Image       => datarobot-<owner>-<env>-AutoScaledWorkerImage

Continue Deletion [y/n]? y
Tue Dec 29 17:55:41 EST 2020 : describe-images
filters:"Name=name,Values=datarobot-<owner>-<env>-AutoScaledWorkerImage" query:'Images[*].{ID:ImageId}'
Tue Dec 29 17:55:43 EST 2020 : deregister-image image-id:<some ami id>
Tue Dec 29 17:55:45 EST 2020 : delete-alarms: datarobot-<owner>-<env>-ScaleUp-Modeling-Capacity-Alarm
Tue Dec 29 17:55:46 EST 2020 : delete-policy: datarobot-<owner>-<env>-ScaleUp-Policy
Tue Dec 29 17:55:47 EST 2020 : delete-alarms: datarobot-<owner>-<env>-ScaleDown-Modeling-Capacity-Alarm
Tue Dec 29 17:55:48 EST 2020 : delete-policy: datarobot-<owner>-<env>-ScaleDown-Policy
Tue Dec 29 17:55:49 EST 2020 : delete-auto-scaling-group: datarobot-<owner>-<env>-AutoScaling-Group
Tue Dec 29 17:55:50 EST 2020 : delete-launch-configuration: datarobot-<owner>-<env>-AutoScaling-LC
```

Where **env** = poc or preprod  
**Owner** = the owner of this cluster