



AI Based Enhancements to Proctoring using Briso – System Manual **ExamsOnline**

Briso-Exams: Version 2.0.0

Document Version History

Author	Version	Status*	Signoff Owner	Date of Update
Anubhav Tiwari	0.1	Ongoing	Nishant Varshney	05/06/2020
Nishant Varshney	0.2			
Nishant Varshney	2.0	Finished	Nishant Varshney	04/07/2020

*Status – Draft, In Review, Signed Off, Revised (Changed post sign-off)

Table of Content

1	Overview	3
2	System Description	4
3	Setup instructions	7
3.1	Server	7
3.1.1	Perquisites	7
3.1.2	Setting up the server	7
3.1.3	Setting up the webserver	9
3.1.4	Auto Scaling Setup.....	10
3.2	Client SDK	11
4	Trouble Shooting	12
4.1	Checking if the Server is running.....	12
4.2	Analyzing the Logs.....	12
4.3	License Expiry	13
4.4	CORS.....	13
4.5	Starting Dev-Test, Uat OR Perf Env	13
	Appendix – Misc. Information	15

1 OVERVIEW

AI based system has been developed by RMgX based on its proprietary Briso product for Examonline to integrate with their online proctoring system to provide advanced AI based monitoring to avoid cheating during the exams and improve their credibility.

The document provides the information about the **Briso-Exam AI** monitoring system – how it functions, its installation, configuration and general troubleshooting.

The document also contains the flow of the process to show the intercommunication between the client-side JavaScript SDK and the server-side SDK.

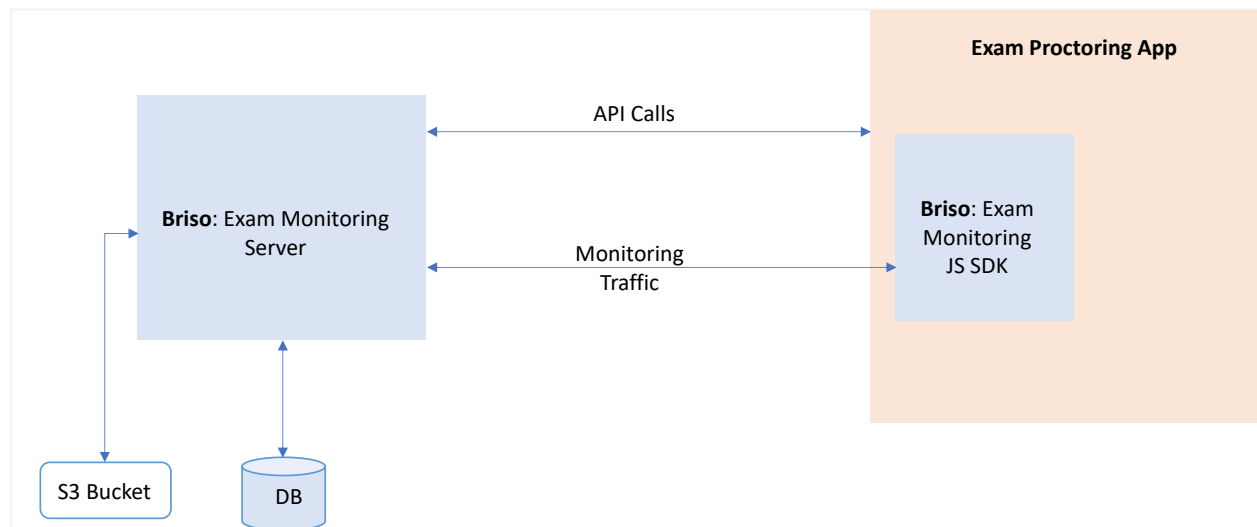
The document contains the instructions on how to setup the server and the client-side SDKs in case of reconfiguration. There are detailed diagrams and commands enclosed to ease the process of setting up the SDKs without requiring any additional support.

The document also contains possible errors that might be encountered while running the product and how to trouble shoot in case of coming across one.

It is emphasized that this document is for 'Examonline' only and is part of the license issued to Examonline for using the platform with limitations established by the license over reuse and redistribution of the source code and resources or any entity related to the beforementioned product. This document cannot be distributed to any other company or shared with unintended parties as it will constitute the breach of licensing terms.

2 SYSTEM DESCRIPTION

Briso Exam monitoring system runs in client-server model with the client and the server intercommunicating with each other for each prediction state. The server handles both the identification phase as well as the monitoring phase with some operations taking place on the client side. The following diagram descriptions how it functions:



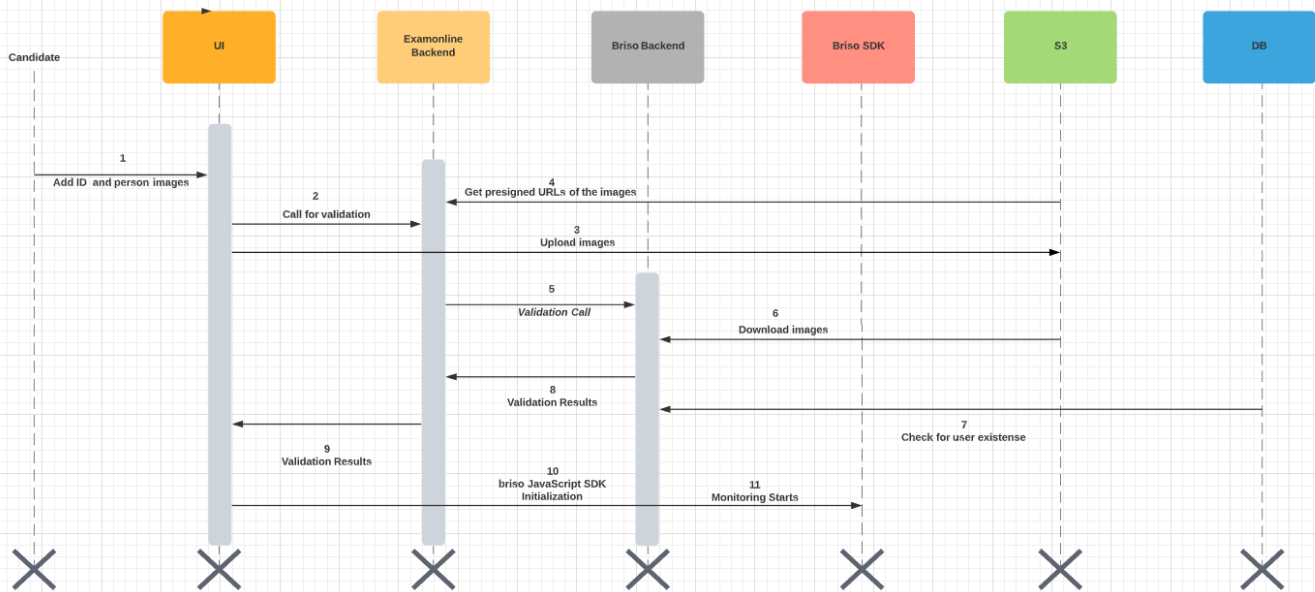
The Briso: Exam Monitoring Server exposes REST based api's namely:

- **register:** To register a new candidate in the system who can then be subsequently monitored
- **verify:** To auto verify the candidate. This API can be used during authorization process to auto authorized the users This APIs allow for automated authentication of a candidate from his/her live captured image from the webcam as well as the image of his/her identification document.
- **update:** To update the candidate details

For running monitoring process, the client and the server maintain a socket connection with the client feeding the server a frame every few seconds and the server logging and emitting in case of any anomaly encountered in the frame. The emitted error is received at the client side which can be used as per the requirements like alerting the proctor about some suspicious activity at the candidate end.

The following diagram depicts the sequence of activities and interaction of various parts of the system to fulfill the various use cases. The depiction uses the standard sequence diagram approach where various parts are lined up as layers and interaction between them is shown using activation timelines and vectored arrows to show the flow of data.

Briso Integration With Examonline Workflow



The following is the detailed information about the API's and their definition

/register			
POST			
Parameter Name	Type	Description	Requirement
token	string	API Key Provided From Briso Server	required
user_id	string	Unique Identifier Of The User	required
user_name	string	Name Of The User	required
user_image_url	string	URL For The Image Of User To Calculate Embeddings	required
user_id_number	string	Identification Document Number For OCR Verification Process	required
distinguisher_id	string	Unique Identifier Of The Exam Authority	required
/update			
POST			
token	string	API Key Provided From Briso Server	required
fields	json	The Key Value Pairs For The Fields To Update, Allowed Update Fields Are : user_name, user_id_number, user_image_url	required, with each updatable key optional.
distinguisher_id	string	Unique Identifier Of The Exam Authority	required
user_id	string	Unique Identifier Of The User, Same As That When Registered.	required
/verify			
POST			
token	string	API Key Provided From Briso Server	required
user_image_url	string	URL For The Image Of User To Calculate Embeddings	required
distinguisher_id	string	Unique Identifier Of The Exam Authority	required
user_id	string	Unique Identifier Of The User, Same As That When Registered.	required
ocr_process	boolean	Whether To Perform OCR Verification Of The ID	required
user_id_image_url	string	The URL For The Image Of The ID Document	optional, required when ocr_process is set to true.

3 SETUP INSTRUCTIONS

3.1 SERVER

Briso AI Server can work on any OS but it is preferable to run it on 'Linux' based environments as they are performant for such types of workloads

3.1.1 Perquisites

The following base software's are required for the application to work, ensure they are installed before proceeding ahead:

- [python3](#) with pip3. See [this link](#) for a step by step guide for doing so on ubuntu server.
Note: Ubuntu 18.04 comes with python3 pre-installed.
- Database: MySQL and SQL Server Database are supported
- [Redis](#). See [this link](#) for a step by step guide on setting it up.
- If the underlying database is MS SQL then ODBC drivers have to be setup, which can be done using the following commands:

```
> sudo su
> curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -
> curl https://packages.microsoft.com/config/ubuntu/18.04/prod.list
> /etc/apt/sources.list.d/mssql-release.list
> sudo apt-get update
> sudo ACCEPT_EULA=Y apt-get install msodbcsql17
> sudo apt-get install unixodbc-dev
```

3.1.2 Setting up the server

It is assumed that Briso Server application is available with you. The following assumes that server will be installed in '/opt/examonline-ai-server' directory

1. At the root of the folder run the command 'pip3 install -r requirements.txt'. This will install the necessary python dependencies required for the program to run
2. Edit the Configuration file located under the 'settings' folder for the environment on which this instance of the application is supposed to run. The settings are already defined, but in case change is required, here is a quick reference to understand the settings and their meaning


```
{
  "ENV_NAME": "<Name Of The Environment To Run The Server On>",
  "USER_NAME": "<User Name Of The Database Connection>",
  "PASSWORD": "<Password For The Database Connection>",
  "DATA_BASE": "<Name Of The Database>",
  "SCHEMA": "<Name Of The Schema (required in case of SQL Server)>",
  "DEPLOYMENT_MODE": "<Leave At Default i.e., 'cloud'>",
  "CLOUD_HOST": "<Host ARN or Endpoint Of The Database>",
  "SERVER_PORT": "<Port On Which To Run Server>",
  "REDIS_PASSWORD": "<Password For Redis Server>",
  "DEBUG": "<Boolean, Whether To Print Debug Logs or Not>",
  "LOGGING_LEVEL": "<Check log_levels.txt>",
  "SECRET_KEY": "<Leave At Default>",
  "SQL_TYPE": "<The Type Of Database Server, 'server' In Case Of SQL Server 'mysql' In Case Of MySQL>",
  "FRAME_S3_ACCESS_KEY": "<Access Key For Your S3 Storage For Storing Snapshots>",
  "FRAME_S3_SECRET_KEY": "<Secret Key For Your S3 Storage For Storing Snapshots>",
  "FRAME_S3_REGION": "<Region For S3 Bucket For Storing Snapshots>",
  "FRAME_S3_BUCKET": "<Name Of The S3 Bucket>",
  "FRAME_S3_BUCKET_ENDPT": "<Folder Endpoint In The Bucket Where To Store The Snapshots>",
  "BRISO_S3_ACCESS_KEY": "<Leave As Default>",
  "BRISO_S3_SECRET_KEY": "<Leave As Default>",
  "BRISO_S3_REGION": "<Leave As Default>",
  "BRISO_LICENSE": "<Leave As Default>"
}
```

3. Now, copy the file "briso-server.service" (available in the root of the package) to "/etc/system.d/system"
4. Run command 'sudo systemctl daemon-reload' to load the service
5. Once service has been corrected installed the following commands can be used to interact with the service. The Briso's Monitoring Server service by defaults runs on 5000 port.

```
> sudo systemctl start briso-server - To start the servers
> sudo systemctl status briso-server - To check server status
> sudo systemctl restart briso-server - To restart the servers
> sudo systemctl stop briso-server - To stop the servers
```

6. To run the server without configuring the service, the following command can be used

```
> nohup python3 server.py 'env_name' &> log.out & echo $! > run.pid
```

Here env_name is the name of the config file without the extension, example, to run in uat mode, config file is uat.json and env_name is uat.

The above command will run the process in the background and log the output in log.out file

3.1.3 Setting up the webserver

The server can run on its own to run in a production scenario it will have to be behind a web proxy. Apache is a good choice and system has been extensively tested against it. The following are the steps for setting up apache webserver:

1. Run the following commands to install apache and necessary modules:

```
sudo apt-get install apache2
sudo a2enmod rewrite headers alias proxy proxy_http
```

2. Once apache is running, update the '000-default.conf' (the default virtual host) located under '/etc/apache/sites-available' with the following :

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost

    # To redirect, HTTP to HTTPS
    RewriteEngine On
    RewriteCond %{HTTP:X-Forwarded-Proto} =http
    RewriteRule . https://%{HTTP:Host}%{REQUEST_URI} [L,R=permanent]

    # CORS Support
    SetEnvIf Origin
"http(s)?://(www\.)?(localhost:4200|uat.examonline.in|assessment.examonline.i
n|examonline.in)$" AccessControlAllowOrigin=$0
    Header set Access-Control-Allow-Origin %{AccessControlAllowOrigin}e
    env=AccessControlAllowOrigin
    Header set Access-Control-Max-Age "600"
    Header set Access-Control-Allow-Credentials "true"

    # Proxy Requests
    ProxyPass /error/ !
    ProxyPass /demo/ !

    ProxyPass / http://127.0.0.1:5000/
    ProxyPassReverse / http://127.0.0.1:5000/

    ErrorDocument 404 /error/error404.html
    ErrorDocument 403 /error/error403.html
    ErrorDocument 500 /error/error500.html
    ErrorDocument 503 /error/error503.html

    DocumentRoot /var/www/html/ai

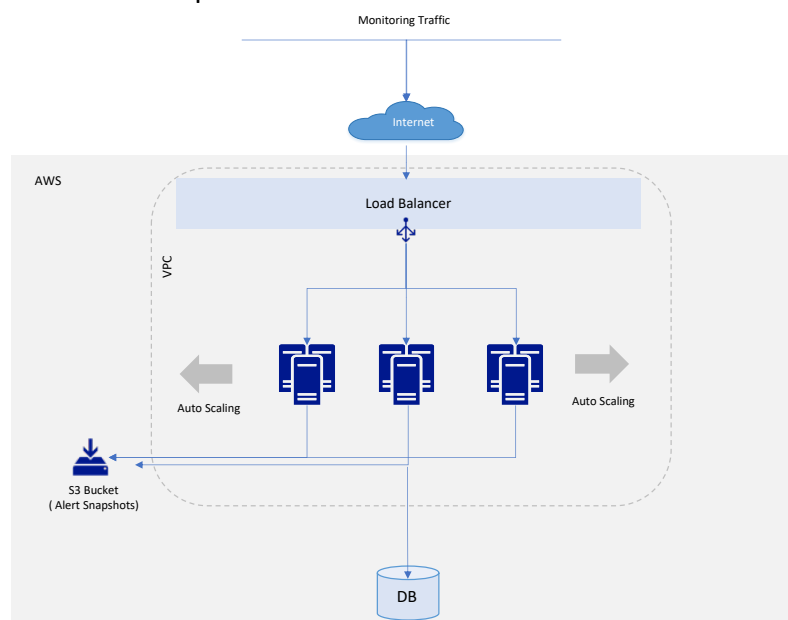
    <Directory /var/www/html/ai>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride All
        Order allow,deny
        allow from all
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

3. Restart apache and visit the server url to view the default page
4. **Note** - Above assumes the Briso Server is running on 5000 port, if in the configuration port has been change, it has to be reflected correctly in the virtual host's proxy configuration, otherwise webserver will not connect with the Briso Server

3.1.4 Auto Scaling Setup

Briso Server is a scalable system, but in production environment to create redundancy and also to allow for scaling of the system along with increment of load, Briso Server can be setup in auto scaling mode as well. The following diagram depicts the auto scaling setup and the various components required to set it up



The following things should be noted, while setting the auto scaling group:

1. Load Balancer should be configured with 'sticky sessions on'. The reason for this is that Briso Client SDK communicates with the Briso Server over socket protocol which is long polling in nature and also not cluster safe. So, a connection made to server should continue to have the traffic from the same client. Briso contains the logic and handling that in case the node handling traffic goes down, the system automatically re-initializes connection on the other node without loss of information and stopping of monitoring session
2. Scaling policy should be configured to spawn new node upon CPU utilization > 70 %. Given the nature of AI computation happening in Briso Server if the CPU goes beyond 75% the results of analysis lose their precision
3. Briso Server requires around 40 secs to start, warm up time of nodes in the scaling policy should be configured keep this in mind.

On Examonline AWS environment, autoscaling groups for each environment type has been fully configured for usage

<input type="checkbox"/>	Name	Launch Configuration / ~	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grac
<input type="checkbox"/>	AI-Live-AutoScaling-Group	AI-EC2-AutoScaling-La...	1	1	1	2	ap-south-1c, ap-south-1b, ap...	150	10
<input type="checkbox"/>	AI-Live-Verification-AutoScaling-Group	AI-EC2-AutoScaling-La...	1	1	1	2	ap-south-1c, ap-south-1b, ap...	150	10
<input type="checkbox"/>	AI-Perf-AutoScaling-Group	AI-EC2-AutoScaling-La...	0	0	0	2	ap-south-1c, ap-south-1b, ap...	10	20
<input type="checkbox"/>	AI-Test-AutoScaling-Group	AI-EC2-AutoScaling-La...	0	0	0	1	ap-south-1c, ap-south-1b, ap...	10	20
<input type="checkbox"/>	AI-UAT-AutoScaling-Group	AI-EC2-AutoScaling-La...	0	0	0	1	ap-south-1c, ap-south-1b, ap...	10	20

The instance count for Perf, Test, UAT environments is kept to 0, to avoid incurring cost, but whenever required, it can be updated via the AWS console to automatically spawn new instances.

3.2 CLIENT SDK

The JS Based SDK has been made available via RMgX's google drive dedicated for Examonline. The various versions are filed under their respective folder.

My Drive > RMgX_Client_Share > ExamsOnline > Briso-SDK

Name	Owner	Last modified	File size
archived	me	23:48 me	—
v1.14.0	me	23:49 me	—
v1.14.0-PATCH	me	23:49 me	—
v2.0.0	me	23:49 me	—

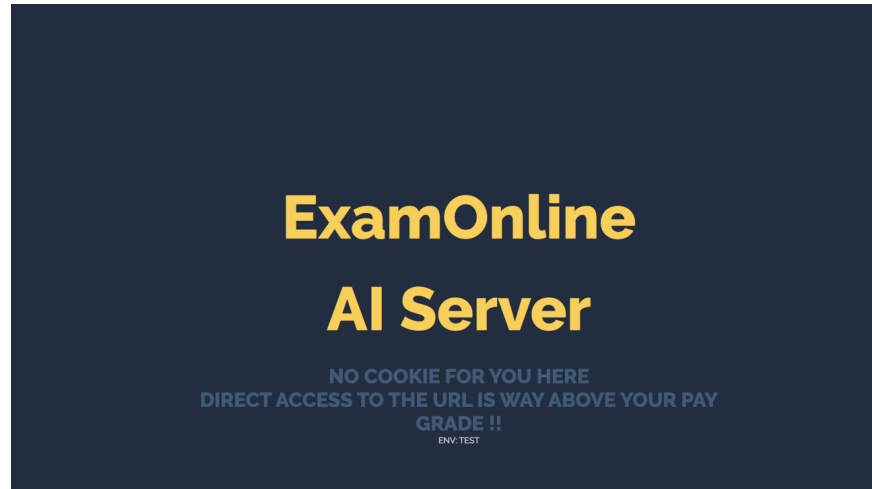
The Client SDK version ties to the backend server and it has to match perfectly, so a mismatched Client SDK version will not work with the backend running on different version. Care must be taken to use the appropriate version matching the server version on which the client will connect.

The integration instructions are included with the SDK code, and it should be referenced for carrying out Client Integration.

4 TROUBLE SHOOTING

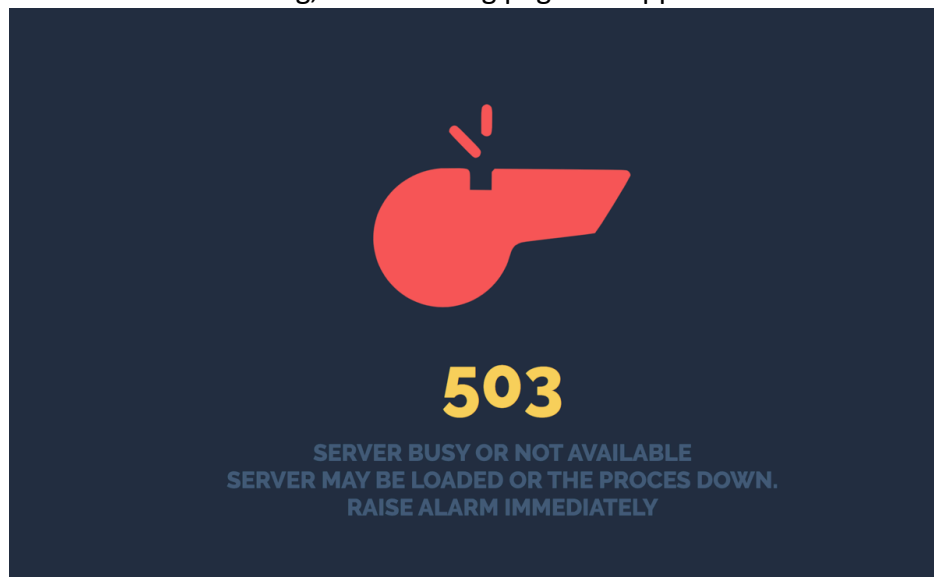
4.1 CHECKING IF THE SERVER IS RUNNING.

Hit the server on its url and the following page should appear:



The page shows the name of the environment configuration use to start it at the bottom of the page. It should be noted and ensured that the value is as intended, else, configuration should be fixed, and server restarted.

If the server is down or not running, the following page will appear



Unless it is intentional, appearance of above page will require, starting the Briso Server Service in the background

4.2 ANALYZING THE LOGS

The application creates the logs which are located at ``logs/app.log`` . One should always view the logs to perform basic troubleshooting

The amount of details in the logs is controlled via the environment variable 'BRISO_DEBUG' . if the value is set to be 'true', then system will generate very verbose logs which can help in investigation of an issue but setting it to be 'true' is not recommended in the scenario of heavy traffic or on production as it slows the execution power of the server.

System is configured with automatic logs roll over once the size of a log file goes beyond a threshold and last 14 log files are retained and rest automatically purged. In case the logs need to be preserved for future analysis, setup should be carried out to move them to a different location

4.3 LICENSE EXPIRY

The system has been configured with a license configured for Examonline based on agreement with RMgX.

If one encounters any errors in the logs like "Wrong Key" or any Errors related to issues while loading models, please contact RMgX Technologies LLP as in such scenario system will not function as intended.

4.4 CORS

The current configuration allows the server access from limited hosts. If more hosts are to be allowed to access the server via the briso client sdk, apaches virtual host configuration should be updated to allow the necessary hosts. See section 3.1.3 for more details.

It should be understood CORS is not an error but a security mechanism, refer to this [link](#) to know more about CORS and its usage.

4.5 STARTING DEV-TEST, UAT OR PERF ENV

To save on cost, the Dev-Test, UAT and Perf environment's autoscaling groups have been configured to spawn 0 nodes. Whenever there is a requirement to use those servers, the autoscaling configuration min server's configuration can be tweaked and it will spawn the necessary nodes to have the application work

Create Auto Scaling group

Actions

Filter:

<input type="checkbox"/>	Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grac
<input type="checkbox"/>	AI-UAT-AutoS...	AI-EC2-AutoScaling-La...	0	0	0	1	ap-south-1c, ap-south-1b, ap...	10	20
<input checked="" type="checkbox"/>	AI-Test-AutoSc...	AI-EC2-AutoScaling-La...	0	0	0	1	ap-south-1c, ap-south-1b, ap...	10	20
<input type="checkbox"/>	AI-Live-Verific...	AI-EC2-AutoScaling-La...	1	1	1	2	ap-south-1c, ap-south-1b, ap...	150	10
<input type="checkbox"/>	ASG-RECOR...	LC-RECORD-PROD-2	1	1	1	4	ap-south-1c, ap-south-1b, ap...	300	300
<input type="checkbox"/>	ASG-JANUS...	LC-JANUS-PROD-2	2	2	2	6	ap-south-1c, ap-south-1b, ap...	300	300
<input type="checkbox"/>	AI-Live-AutoSc...	AI-EC2-AutoScaling-La...	1	1	1	2	ap-south-1c, ap-south-1b, ap...	150	10

Edit details - AI-Test-AutoScaling-Group

×

☐ Launch Configuration

Launch Template

AI-EC2-AutoScaling-Launch-Template-Test

▼

Launch Template Version

Latest

▼

Launch Template Description

-

Fleet Composition

☒ Adhere to the launch template

The launch template determines the instance type and purchase option (On-Demand or Spot).

☐ Combine purchase options and instances

Choose a mix of On-Demand Instances and Spot Instances and multiple instance types. Spot Instances are automatically launched at the lowest price available.

Desired Capacity

0

Min

0

Max

1

Availability Zone(s)

ap-south-1a

×

ap-south-1b

×

ap-south-1c

×