EPAM Syndicate RightSizer

User Guide

May 2024

R8-01

Version 1.0

Legal Notice: This document is property of EPAM and may not be disclosed, distributed, or reproduced without the prior written permission of EPAM®.

CONTENT

PREFACE

ABOUT THIS GUIDE

The purpose of this Guide is to provide the basic information about Syndicate RightSizer tool, its purpose, installation and main tools, as well as provides the link to the tool repository.

AUDIENCE

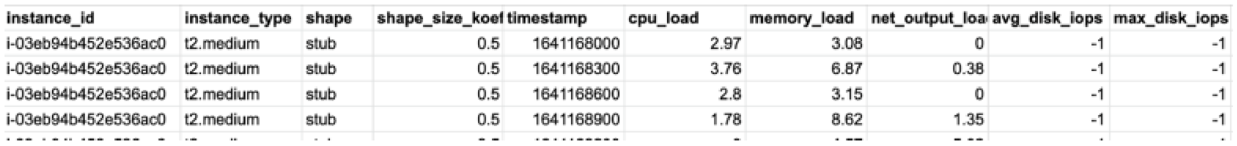
The guide is designed for Syndicate RightSizer users who are willing to perform virtual instances review and get rightsizing recommendations.

1 INTRODUCTION TO EPAM SYNDICATE RIGHTSIZER

EPAM Syndicate Rightsizer (also referred as r8s) is an ML-based mechanism aimed to create suggestions on the specific virtual instance's optimization, based on their performance data. As input data, the Rightsizer receives the data about the instance's configuration, timelines, and performance. Once the input data is processed, the tool generates recommendations, which suggest:

- Specific resources shutdown (termination).
- Start/Stop schedules for cost optimization of the specific resources.
- Scale up recommendation for the resources that are currently overloaded.
- Scale down recommendations for the resources that are underutilized.
- Split an instance load among multiple instances of different configurations.

The collected metrics file has the structure as on the screenshot below:



instance_id	instance_type	shape	shape_size_koef	timestamp	cpu_load	memory_load	net_output_loa	avg_disk_iops	max_disk_iops
i-03eb94b452e536ac0	t2.medium	stub	0.5	1641168000	2.97	3.08	0	-1	-1
i-03eb94b452e536ac0	t2.medium	stub	0.5	1641168300	3.76	6.87	0.38	-1	-1
i-03eb94b452e536ac0	t2.medium	stub	0.5	1641168600	2.8	3.15	0	-1	-1
i-03eb94b452e536ac0	t2.medium	stub	0.5	1641168900	1.78	8.62	1.35	-1	-1

Each result file includes the per-instance recommendations, each containing the following keys:

- resource_id – the ID of the resource for which the recommendations are created.
- resource_type – the type of resource for which the recommendations are created.
- source – the tool that generated the recommendation (in such case 'RIGHTSIZER').
- severity – the finding severity level; 'MEDIUM' is used for the Rightsizer tool.
- schedule – the suggested start/stop pattern for the instance.
- recommended_shapes – the suggested scale up/down configuration for the instance.
- savings - calculated savings for proposed r8s recommendations.
- advanced - detailed statistical information about the processed dataset.
- stats – the additional information about instance processing.
- meta (duplicate meta passed by Maestro) – meta information retrieved as an input, including recommendations from other sources.
- general_action – the generalized summary for the action recommended for the instance.

The 'general_action' section stores the generalized recommended action for the given instance. It should be used to define which part of the result should be referred to.

The possible values are:

- ERROR – displayed if some error occurred while processing instance. Referring to stats will provide more detailed error message.
- NO_ACTION – identifies that the instance is well-optimized, no actions needed.
- SHUTDOWN – the instance can be shut down.
- SCHEDULE – a schedule can be applied to the instance.
- SCALE_UP – changing shape to a larger one is recommended.
- SCALE_DOWN – changing shape to a smaller one is recommended.
- CHANGE_SHAPE – change shape to a different one is recommended (for example, use another family with same CPU/RAM, or increasing the CPU while decreasing a RAM).

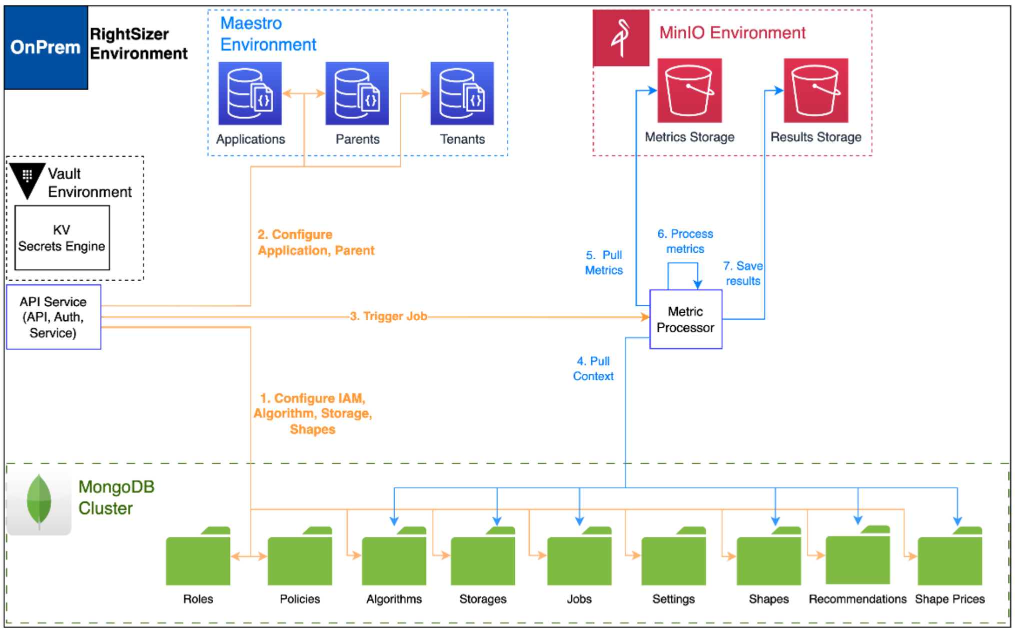
- SPLIT – consider dividing instance into several separate instances with specified shapes (has load-dependent recommendations).
- MULTIPLE - consider choosing between reshaping and scheduling.

2 EPAM SYNDICATE RIGHTSIZER SETUP

Below you can find the details of the RightSizer configuration as well as the details of r8s CLI tool installation process.

Configuration details:

- r8s API HOST – sets during RightSizer configuration.
- r8s API user - configured during RightSizer configuration.
- Configured RightSizer for one of the available options: tenant/several tenants/all tenants in cloud/several clouds/all available clouds.



RightSizer configuration includes the following steps:

- Set up an S3 bucket with read/write access to it for storing metrics and reports should be created before Application configuration.
- Verify that an active Application with ‘RIGHTSIZER’ type exists. The application status can be checked in DB (based on installation type - SaaS or On-Prem) in the ‘Applications’ collection or via the r8s CLI command: ‘r8s application describe --json'.
- The RightSizer algorithm should be added for each cloud. To make sure that algorithm exists, execute the following command in the r8s cli: ‘r8s algorithm describe --json' or check it in the Mongo ‘r8s’ database in the ‘algorithm’ collection.
- Verify that the RightSizer license for each cloud exists. The licenses can be checked in the DB (based on installation type - SaaS or On-Prem) in the ‘CSLMLicense’ collection where service_type = RIGHTSIZER.
- Verify that active Applications with ‘RIGHTSIZER_LICENSES’ type exist. The applications status can be checked in DB (based on installation type - SaaS or On-Prem) in the ‘Applications’ collection

where type is RIGHTSIZER_LICENSES or via the r8s CLI command: 'r8s application licenses describe --json'.

- Verify that an active Parent with 'Rightsizer' type exists. The Parent is auto generated and adds during host application creation. The Parent links host application to tenants.
- Verify that an active Parents with the type 'RIGHTSIZER_LICENSES' exist. The parent status can be checked in DB (based on installation type - SaaS or On-Prem) in the 'Parent' collection with the type 'RIGHTSIZER_LICENSES' or via the r8s cli command: 'r8s parent describe --json'. Such parent indicates that license is activated for tenant(s).
- Shapes should be added for each cloud. To make sure that shapes exist, execute the following command in the r8s cli: 'r8s shape describe --json' or check it in the Mongo 'r8s' database in the 'shape' collection.
- Shapes prices should be added for each cloud and each region. To make sure that shape prices exist, execute the following command in the r8s cli: 'r8s shape price describe --json' or check it in the Mongo 'r8s' database in the 'shape_price' collection.
- The following setting 'MANAGEMENT_RECOMMENDATIONS_SOURCES_MAP' should be enabled to display recommendations on UI on the Management page.

2.1 PREREQUISITES FOR R8S CLI

- Python 3 Installed.
- Pip installed.
- r8s repository https://github.com/epam/r8s is located.
- virtualenv installed (NOTE: Due to the best Python practices, it is recommended to use virtual environment to protect against dependency breakage. You can perform r8s cli tool installation without this step. Need to mention that in case you want to install the to the created virtual environment you will need to activate the virtual environment with installed r8s cli before using it).

2.2 R8S CLI INSTALLATION

To install the r8s CLI tool, do the following steps:

1. Clone project: `git clone https://github.com/epam/r8s`
2. Move to project root `cd r8s`
3. Create virtual environment: `virtualenv  -p python venv`
4. Activate virtual environment: `. venv/bin/activate`
5. Install r8s cli: `python -m pip install r8s/`
6. Configure api link: `r8s configure --api_link $API_LINK`
7. Login: `r8s login --username $USERNAME --password $PASSOWORD`

Refer to r8s cli for detailed description of available commands.

3 SYNDICATE RIGHTSIZER TESTING

To evaluate Rightsizer tool quality, the checks are divided for the different approaches:

- Using current load of the virtual instances.
- Using mocked generated datasets (based on specific instance tags).

3.1 CURRENT LOAD OF THE VIRTUAL INSTANCES APPROACH

For the current approach, the QA team uses metrics collected for the specific period for virtual machines. The lifetime duration of instance should be not less than 3 days. The duration can be updated via License Management cli command:

```
cslm algorithm update_recommendation_settings --min_allowed_days $DAYS_COUNT
```

For all types of recommendations, except SCHEDULE, the tested instance lifetime should be no less than 3 days.

In case an instance lifetime is less than 3 days, the recommendation will be ERROR with the following error message: “Insufficient data. Analyzed period must be larger than a full 3 day(s) with 5-min frequency of records.”

For the SCHEDULE recommendation to be created, the instance lifetime should be no less than 14 days.

It can be updated with the following CLI command:

```
cslm algorithm update_recommendation_settings --min_allowed_days_schedule
$DAYS_COUNT
```

Once the previous steps were done, the next actions should be executed:

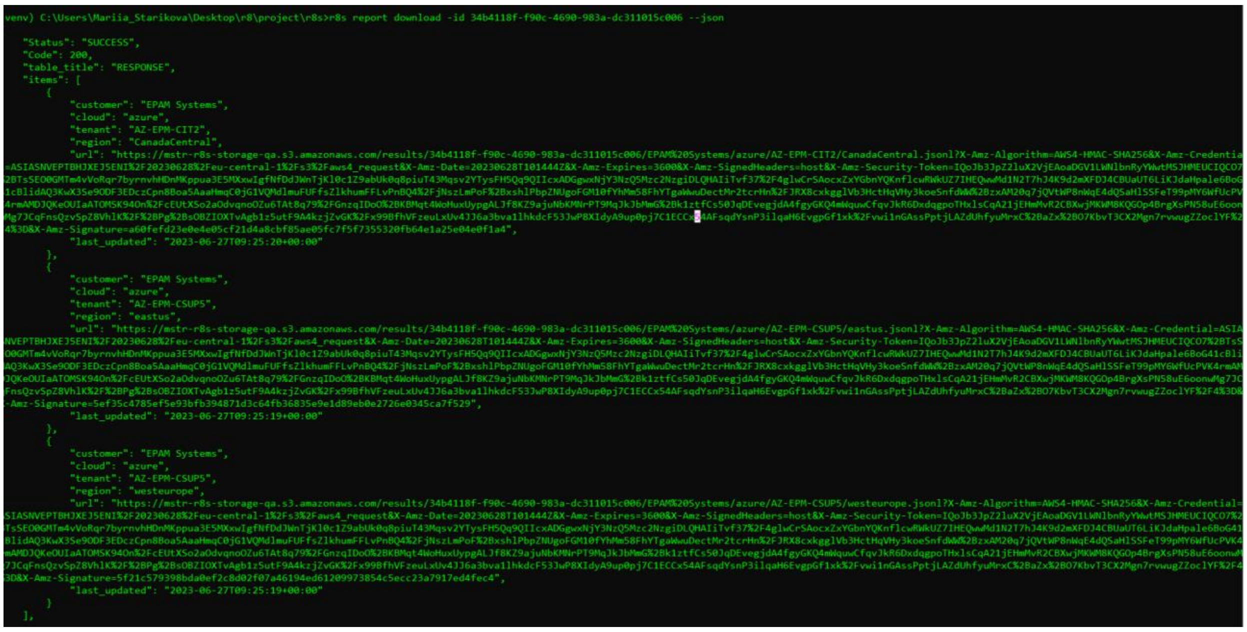
1. To collect metrics for instances in tenants where Rightsizer is activated, execute the ‘Generate Rightsizer recommendations’ job from the JMX page.
   It will gather metrics for instances, create meta file (one for all instances in region) which contains information about instances tags, previous recommendations, schedules, creation timestamps.
   Metrics files are saved to the AWS S3 bucket that was created during Rightsizer configuration.
   The path to the metrics and meta.json file is:
   created bucket -> metrics -> customer -> cloud -> tenant name -> region -> date (previous day).
2. After the metrics are gathered via the ‘Generate Rightsizer recommendations’ job from the JMX page, the Rightsizer jobs are automatically submitted. The Rightsizer job submission can be checked via r8s CLI by the *r8s job submit* command for the RIGHTSIZER_LICENSES application (required). The job submission can be executed with additional parameters such as specific period and/or specific parent and/or specific tenant. Please note: in case the metrics haven’t been updated since the last scan, the metrics won't be downloaded. The job results will include previous instance recommendations.
3. There are several ways to check Rightsizer job execution status, all should be tested:
   - Check job status in AWS Batch (r8s-job-queue).
   - Check status in the ‘r8s’ Mongo database in the ‘job’ collection.
   - Check status in the r8s cli via the ‘r8s job describe’ command.

   In case job status is FAILED there are three ways to check failed reason:
   - Open FAILED job in the AWS Batch and click on the Log stream name (the Cloud Watch will be open).

- Check the fail_reason in the ‘r8s’ Mongo database in the ‘job’ collection.
- Check the fail_reason for job described via r8s cli using the ‘r8s job describe’ command

In case job has SUCCEEDED status the recommendation will save to the created S3 Bucket by path: created bucket -> job id -> customer -> cloud -> tenant name -> region.

4. There are several ways how to download recommendations, all should be tested:
    - Using the r8s cli command: *r8s report download*. The URL from the command response should be used for recommendation downloading. The response can contain several URLs for different tenants in different regions.
    - Download report from s3 bucket by the path created bucket -> job id -> customer -> cloud -> tenant name -> region.



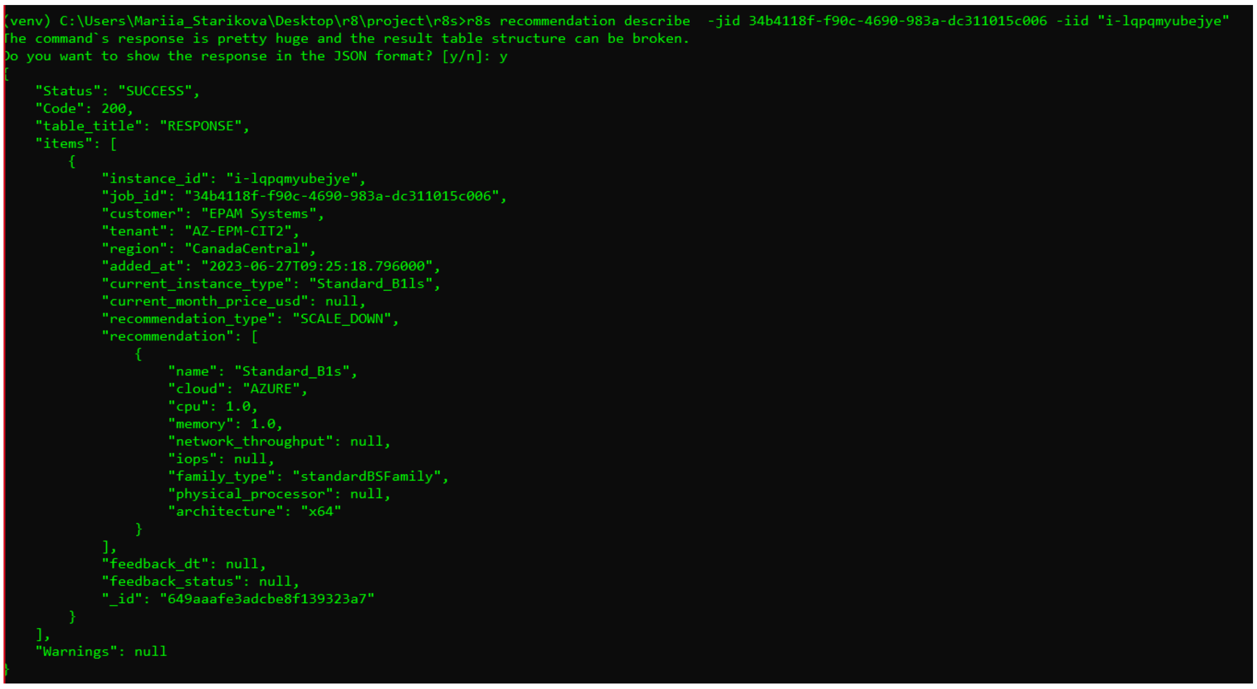
```
venv) C:\Users\Mariia_Starikova\Desktop\r8\project\r8s>r8s report download -id 34b4118f-f90c-4690-983a-dc311015c006 --json

    "Status": "SUCCESS",
    "Code": 200,
    "table_title": "RESPONSE",
    "items": [
        {
            "customer": "EPAM Systems",
            "cloud": "azure",
            "tenant": "AZ-EPM-CIT2",
            "region": "CanadaCentral",
            "url": "https://mstr-r8s-storage-qa.s3.amazonaws.com/results/34b4118f-f90c-4690-983a-dc311015c006/EPAM%20Systems/azure/AZ-EPM-CIT2/CanadaCentral.jsonl?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credentia
=ASIASNVEPTBHJXEJ5ENI%2F20230628%2Feu-central-1%2Fs3%2Faws4_request&X-Amz-Date=20230628T101444Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Security-Token=IQoJb3JpZ2luX2VjEAoaDGV1LWNlbnRyYWwtMSJHMEUCIQCO7
2BTsSEO0GMTm4vVoRqr7byrnvhHDnMKppua3E5MXxwIgfNfDdJWnTjKl0c1Z9abUk0q8piuT43Mqsv2YTysFH5Qq9QIIcxADGgwxNjY3NzQ5Mzc2NzgiDLQHAIiTvf37%2F4glwCrSAocxZxYGbnYQKnflcwRWkUZ7IHEQwwMd1N2T7hJ4K9d2mXFDJ4CBUaUT6LiKJdaHpale6BoG
1cBlidAQ3KwX3Se9ODF3EDczCpn8Boa5AaaHmqC0jG1VQMdlmuFUFfsZlkhumFFLvPnBQ4%2FjNszLmPoF%2BxshlPbpZNUgoFGM10fYhMm58FhYTgaWwuDectMr2tcrHn%2FJRX8cxkgglVb3HctHqVHy3koeSnfdWW%2BzxAM20q7jQVtWP8nWqE4dQSaHlSSFeT99pMY6WfUcPV
4rmAMDJQKeOUIaATOMSK94On%2FcEUtXSo2aOdvqnoOZu6TAt8q79%2FGnzqIDoO%2BKBMqt4WoHuxUypgALJf8KZ9ajuNbKMNrPT9MqJkJbMmG%2Bk1ztfCs50JqDEvegjdA4fgyGKQ4mWquwCfqvJkR6DxdqgpoTHxlsCqA21jEHmMvR2CBXwjMKWM8KQGOp4BrgXsPN58uE6oon
Mg7JCqFnsQzvSpZ8VhlK%2F%2BPg%2BsOBZIOXTvAgb1z5utF9A4kzjZvGK%2Fx99BfhVFzeuLxUv4JJ6a3bva1lhkdcF53JwP8XIdyA9up0pj7C1ECCx54AFsqdYsnP3ilqaH6EvgpGf1xk%2Fvwi1nGAssPptjLAZdUhfyuMrxC%2BaZx%2BO7KbvT3CX2Mgn7rvwugZZoclYF%2
4%3D&X-Amz-Signature=a60fefd23e0e4e05cf21d4a8cbf85ae05fc7f5f7355320fb64e1a25e04e0f1a4",
            "last_updated": "2023-06-27T09:25:20+00:00"
        },
        {
            "customer": "EPAM Systems",
            "cloud": "azure",
            "tenant": "AZ-EPM-CSUP5",
            "region": "eastus",
            "url": "https://mstr-r8s-storage-qa.s3.amazonaws.com/results/34b4118f-f90c-4690-983a-dc311015c006/EPAM%20Systems/azure/AZ-EPM-CSUP5/eastus.jsonl?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=ASIA
NVEPTBHJXEJ5ENI%2F20230628%2Feu-central-1%2Fs3%2Faws4_request&X-Amz-Date=20230628T101444Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Security-Token=IQoJb3JpZ2luX2VjEAoaDGV1LWNlbnRyYWwtMSJHMEUCIQCO7%2BTsS
O0GMTm4vVoRqr7byrnvhHDnMKppua3E5MXxwIgfNfDdJWnTjKl0c1Z9abUk0q8piuT43Mqsv2YTysFH5Qq9QIIcxADGgwxNjY3NzQ5Mzc2NzgiDLQHAIiTvf37%2F4glwCrSAocxZxYGbnYQKnflcwRWkUZ7IHEQwwMd1N2T7hJ4K9d2mXFDJ4CBUaUT6LiKJdaHpale6BoG41cBli
AQ3KwX3Se9ODF3EDczCpn8Boa5AaaHmqC0jG1VQMdlmuFUFfsZlkhumFFLvPnBQ4%2FjNszLmPoF%2BxshlPbpZNUgoFGM10fYhMm58FhYTgaWwuDectMr2tcrHn%2FJRX8cxkgglVb3HctHqVHy3koeSnfdWW%2BzxAM20q7jQVtWP8nWqE4dQSaHlSSFeT99pMY6WfUcPVK4rmAM
JQKeOUIaATOMSK94On%2FcEUtXSo2aOdvqnoOZu6TAt8q79%2FGnzqIDoO%2BKBMqt4WoHuxUypgALJf8KZ9ajuNbKMNrPT9MqJkJbMmG%2Bk1ztfCs50JqDEvegjdA4fgyGKQ4mWquwCfqvJkR6DxdqgpoTHxlsCqA21jEHmMvR2CBXwjMKWM8KQGOp4BrgXsPN58uE6oonwMg7JC
FnsQzvSpZ8VhlK%2F%2BPg%2BsOBZIOXTvAgb1z5utF9A4kzjZvGK%2Fx99BfhVFzeuLxUv4JJ6a3bva1lhkdcF53JwP8XIdyA9up0pj7C1ECCx54AFsqdYsnP3ilqaH6EvgpGf1xk%2Fvwi1nGAssPptjLAZdUhfyuMrxC%2BaZx%2BO7KbvT3CX2Mgn7rvwugZZoclYF%2F4%3D&
-Amz-Signature=5ef35c4785ef5e93bfb394871d3c64fb36835e9e1d89eb0e2726e0345ca7f529",
            "last_updated": "2023-06-27T09:25:19+00:00"
        },
        {
            "customer": "EPAM Systems",
            "cloud": "azure",
            "tenant": "AZ-EPM-CSUP5",
            "region": "westeurope",
            "url": "https://mstr-r8s-storage-qa.s3.amazonaws.com/results/34b4118f-f90c-4690-983a-dc311015c006/EPAM%20Systems/azure/AZ-EPM-CSUP5/westeurope.jsonl?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=
SIASNVEPTBHJXEJ5ENI%2F20230628%2Feu-central-1%2Fs3%2Faws4_request&X-Amz-Date=20230628T101444Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Security-Token=IQoJb3JpZ2luX2VjEAoaDGV1LWNlbnRyYWwtMSJHMEUCIQCO7%2
TsSEO0GMTm4vVoRqr7byrnvhHDnMKppua3E5MXxwIgfNfDdJWnTjKl0c1Z9abUk0q8piuT43Mqsv2YTysFH5Qq9QIIcxADGgwxNjY3NzQ5Mzc2NzgiDLQHAIiTvf37%2F4glwCrSAocxZxYGbnYQKnflcwRWkUZ7IHEQwwMd1N2T7hJ4K9d2mXFDJ4CBUaUT6LiKJdaHpale6BoG41
BlidAQ3KwX3Se9ODF3EDczCpn8Boa5AaaHmqC0jG1VQMdlmuFUFfsZlkhumFFLvPnBQ4%2FjNszLmPoF%2BxshlPbpZNUgoFGM10fYhMm58FhYTgaWwuDectMr2tcrHn%2FJRX8cxkgglVb3HctHqVHy3koeSnfdWW%2BzxAM20q7jQVtWP8nWqE4dQSaHlSSFeT99pMY6WfUcPVK4
mAMDJQKeOUIaATOMSK94On%2FcEUtXSo2aOdvqnoOZu6TAt8q79%2FGnzqIDoO%2BKBMqt4WoHuxUypgALJf8KZ9ajuNbKMNrPT9MqJkJbMmG%2Bk1ztfCs50JqDEvegjdA4fgyGKQ4mWquwCfqvJkR6DxdqgpoTHxlsCqA21jEHmMvR2CBXwjMKWM8KQGOp4BrgXsPN58uE6oonwM
7JCqFnsQzvSpZ8VhlK%2F%2BPg%2BsOBZIOXTvAgb1z5utF9A4kzjZvGK%2Fx99BfhVFzeuLxUv4JJ6a3bva1lhkdcF53JwP8XIdyA9up0pj7C1ECCx54AFsqdYsnP3ilqaH6EvgpGf1xk%2Fvwi1nGAssPptjLAZdUhfyuMrxC%2BaZx%2BO7KbvT3CX2Mgn7rvwugZZoclYF%2F4
3D&X-Amz-Signature=5f21c579398bda0ef2c8d02f07a46194ed61209973854c5ecc23a7917ed4fec4",
            "last_updated": "2023-06-27T09:25:19+00:00"
        }
    ],
```

Figure 1 - Recommendations download

5. There are several ways to check recommendation type that Rightsizer recommend applying to the instance, all should be tested:

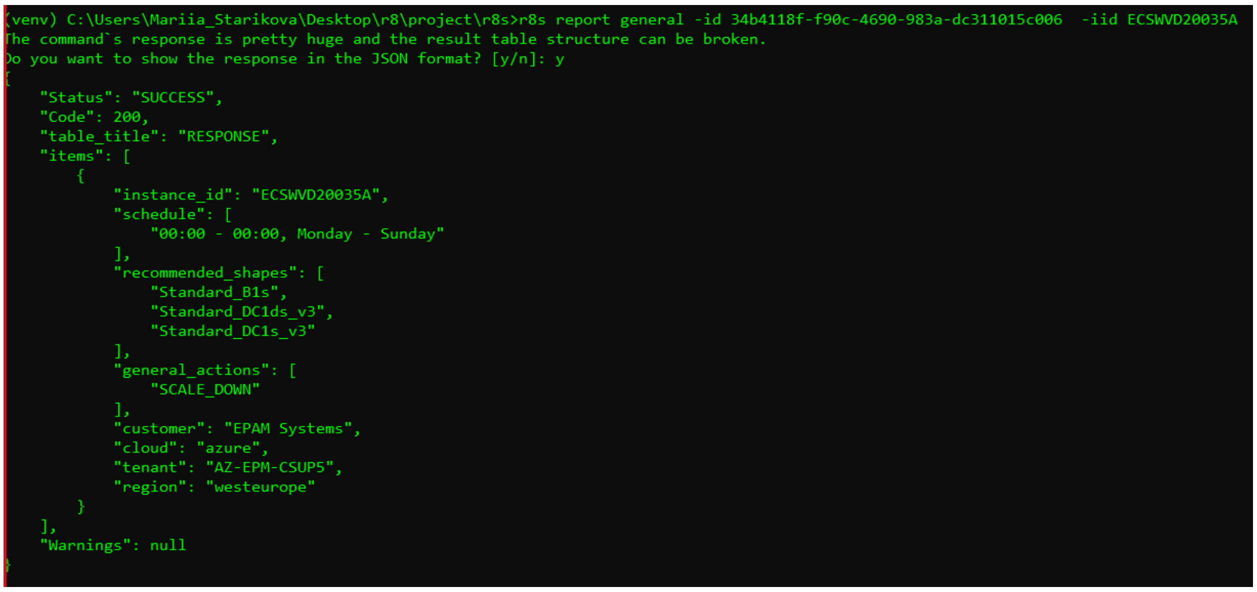
- Using r8s cli command: *r8s recommendations describe*.
  The response contains the main information about the instance (instance id, tenant, region, current shape, customer) and the provided recommendations (recommendation type, recommendation, feedback status, feedback dt, last metric capture date).



```
(venv) C:\Users\Mariia_Starikova\Desktop\r8\project\r8s>r8s recommendation describe  -jid 34b4118f-f90c-4690-983a-dc311015c006 -iid "i-lqpqmyubejye"
The command`s response is pretty huge and the result table structure can be broken.
Do you want to show the response in the JSON format? [y/n]: y
{
    "Status": "SUCCESS",
    "Code": 200,
    "table_title": "RESPONSE",
    "items": [
        {
            "instance_id": "i-lqpqmyubejye",
            "job_id": "34b4118f-f90c-4690-983a-dc311015c006",
            "customer": "EPAM Systems",
            "tenant": "AZ-EPM-CIT2",
            "region": "CanadaCentral",
            "added_at": "2023-06-27T09:25:18.796000",
            "current_instance_type": "Standard_B1ls",
            "current_month_price_usd": null,
            "recommendation_type": "SCALE_DOWN",
            "recommendation": [
                {
                    "name": "Standard_B1s",
                    "cloud": "AZURE",
                    "cpu": 1.0,
                    "memory": 1.0,
                    "network_throughput": null,
                    "iops": null,
                    "family_type": "standardBSFamily",
                    "physical_processor": null,
                    "architecture": "x64"
                }
            ],
            "feedback_dt": null,
            "feedback_status": null,
            "_id": "649aaafe3adcbe8f139323a7"
        }
    ],
    "Warnings": null
}
```

Figure 2 - Recommendations type check in instance details

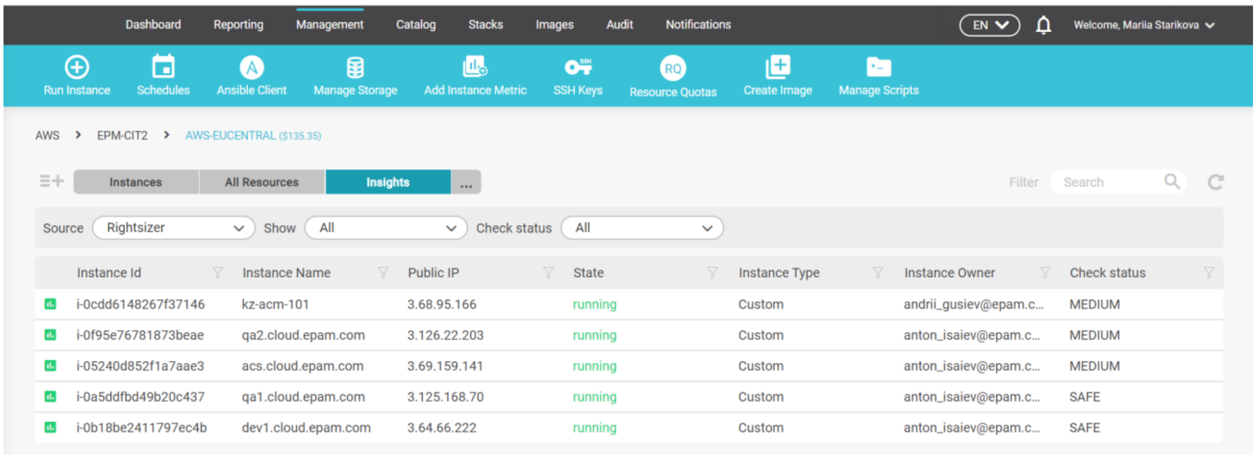
- Using r8s cli command: *r8s report general*.
  The response contains information about general action - recommendation type, schedule, recommended shapes, customer, cloud, region, and tenant. In case the command is executed with --detailed flag, the response contains full content of recommendation – instance meta, stats, savings, full information of recommended shapes. In case the recommendation for the instance is generated for the first time, the ‘advanced’ section should be shown in the response, but if it is not the first recommendation and metrics for the instance are not changed – the section will not be shown.



```
(venv) C:\Users\Mariia_Starikova\Desktop\r8\project\r8s>r8s report general -id 34b4118f-f90c-4690-983a-dc311015c006  -iid ECSWVD20035A
The command`s response is pretty huge and the result table structure can be broken.
Do you want to show the response in the JSON format? [y/n]: y
{
    "Status": "SUCCESS",
    "Code": 200,
    "table_title": "RESPONSE",
    "items": [
        {
            "instance_id": "ECSWVD20035A",
            "schedule": [
                "00:00 - 00:00, Monday - Sunday"
            ],
            "recommended_shapes": [
                "Standard_B1s",
                "Standard_DC1ds_v3",
                "Standard_DC1s_v3"
            ],
            "general_actions": [
                "SCALE_DOWN"
            ],
            "customer": "EPAM Systems",
            "cloud": "azure",
            "tenant": "AZ-EPM-CSUP5",
            "region": "westeurope"
        }
    ],
    "Warnings": null
}
```

Figure 3 - Recommendation type check in general report

- Using Maestro Web UI.
  Navigate to the Management page, Insights mode and choose Rightsizer source. Checking the status in the Instances table will provide the information about the instance state based on the Rightsizer checking.
  Navigate to the Content View to check the recommendation provided by the Rightsizer.



Source: Rightsizer · Show: All · Check status: All

	Instance Id	Instance Name	Public IP	State	Instance Type	Instance Owner	Check status
🟩	i-0cdd6148267f37146	kz-acm-101	3.68.95.166	running	Custom	andrii_gusiev@epam.c...	MEDIUM
🟩	i-0f95e76781873beae	qa2.cloud.epam.com	3.126.22.203	running	Custom	anton_isaiev@epam.c...	MEDIUM
🟩	i-05240d852f1a7aae3	acs.cloud.epam.com	3.69.159.141	running	Custom	anton_isaiev@epam.c...	MEDIUM
🟩	i-0a5ddfbd49b20c437	qa1.cloud.epam.com	3.125.168.70	running	Custom	anton_isaiev@epam.c...	SAFE
🟩	i-0b18be2411797ec4b	dev1.cloud.epam.com	3.64.66.222	running	Custom	anton_isaiev@epam.c...	SAFE

Figure 4 - Recommendations on Maestro UI

6. Analyze the provided recommendations. Gathered metrics are used to analyze the recommendation. The main goal of recommendation analysis is to define if the recommended action is acceptable for the provided instance. To evaluate acceptance of a provided recommendation it should be compared with cloud native console recommendation (AWS Trusted Adviser, Google Recommender, Azure Advisor). After the analysis is performed, the QA Engineer (user) decides if the provided recommendation will be applied or ignored.
7. **Apply recommendation flow:** In case of agreement with the provided recommendation, the recommended actions are performed to the instance.
   There are two ways to respond to recommendations:
   - Using '*r8s recommendation update*' cli command with --feedback_status parameter and APPLIED value in it. Executing the command does not mean applying the recommendation to instance, it means that the feedback is provided for existing recommendation. And the next Rightsizer recommendations will be based on the current feedback.
   - Using Maestro Web UI -> Management page -> Insights mode -> Rightsizer source. Open the Content View for the instance and click on the 'Fix it' button.
8. **Ignore recommendation flow**: In case of disagreement with the provided recommendation, there are two ways of response:
   - Using 'r8s recommendation update' cli command with --feedback_status parameter. It is possible to respond with the following statuses: DONT_RECOMMEND, WRONG, TOO_LARGE, TOO_SMALL, TOO_EXPENSIVE.
     - DONT_RECOMMEND means that the user does not want to apply this recommendation and the next time Rightsizer will not recommend this action.
     - WRONG means that the recommendation is not applicable for such instance and the next time Rightsizer will not recommend this specific shape/schedule.

- TOO_LARGE means that the recommended shapes are too large for the specified tenant and cannot be used for the instance. The next time Rightsizer will recommend smaller shapes.
- TOO_SMALL means that the recommended shapes are too small based on user analyze and plans of instance usage.
- TOO_EXPENSIVE means that the recommended action will result in increasing project costs.

- Using Maestro Web UI -> Management page -> Insights mode -> Rightsizer source. Open the Content View for the instance and click on the 'Ignore' button. After clicking the button, the recommendation with such type will be postponed for 30 days. To update the type of recommendations or the period to postpone, the Recommendations Settings wizard should be used (Dashboard page or In-place wizard on the Management page when the Insights mode is open and Rightsizer source is selected).

3.2 MOCKED GENERATED DATASETS BASED ON SPECIFIC INSTANCE TAGS APPROACH.

For the current approach, the QA team uses a predefined dataset that is based on specific instance tags. Moreover, the metrics for an instance should be gathered beforehand.

This approach replaces actual instance metrics with generated datasets by specifying instance tags. To apply this way of testing, the following tag should be set to the instance:

- `r8s_test_case`[REQUIRED] - defines a default config that will be used to generate metric files. Supported values: `NO_ACTION`, `SCALE_DOWN`, `SCALE_UP`, `SHUTDOWN`, `SPLIT`, 'SCHEDULE`.

To expand test coverage, the following additional tags can be set to the instance during testing:

- `r8s_period_days` - defines a time period of generated metrics file
- `r8s_cpu_load` - defines a desired average CPU load
- `r8s_memory_load` - defines a desired average memory load
- `r8s_avg_disk_iops` - defines a desired average disk iops load
- `r8s_max_disk_iops` - defines a desired max disk iops load
- `r8s_net_output_load` - defines a desired average net output load
- `r8s_std` - defines a deviation that will be used to randomize loads

For the `SCHEDULE` type the specific tags should be set:

- `r8s_cron_start` - cron expression that handles instance starts
- `r8s_cron_stop` - cron expression that handles instance stops

For the `SPLIT` type the specific tags should be set:

- `r8s_cpu_load` - defines a desired average CPU load *for several types of loads*

- `r8s_memory_load` - defines a desired average memory load *for several types of loads*
- `r8s_probability` - defines a covered percentage for each type of load

Without additional tags, the resulting recommendation will have the same general_action as the tag value.

For testing purpose, the QA Team uses predefined configs for the tags which are presented bellow for each type:

```
 "SCALE_DOWN": {
        "r8s_period_days": 14,
        "r8s_cpu_load": 20,
        "r8s_memory_load": 20,
        "r8s_avg_disk_iops": -1,
        "r8s_max_disk_iops": -1,
        "r8s_net_output_load": -1,
        "r8s_std": 2,
    },
    "SCALE_UP": {
        "r8s_period_days": 14,
        "r8s_cpu_load": 80,
        "r8s_memory_load": 80,
        "r8s_avg_disk_iops": -1,
        "r8s_max_disk_iops": -1,
        "r8s_net_output_load": -1,
        "r8s_std": 2,
    },
    "EMPTY": {
        "r8s_period_days": 14,
        "r8s_cpu_load": 45,
        "r8s_memory_load": 55,
        "r8s_avg_disk_iops": -1,
        "r8s_max_disk_iops": -1,
        "r8s_net_output_load": -1,
        "r8s_std": 2,
    },
    "SHUTDOWN": {
        "r8s_period_days": 14,
        "r8s_cpu_load": 5, // means load in scheduled period.
        "r8s_memory_load": 3, // means load in scheduled period
        "r8s_avg_disk_iops": -1,
        "r8s_max_disk_iops": -1,
        "r8s_net_output_load": -1,
        "r8s_std": 1,
    },
    "SPLIT": {
        "r8s_period_days": 14,
        "r8s_cpu_load": [25, 80], // means different load for each load
types
        "r8s_memory_load": [25, 80], // means different load for each load
types
        "r8s_avg_disk_iops": -1,
        "r8s_max_disk_iops": -1,
        "r8s_net_output_load": -1,
        "r8s_probability": [50, 50], // means % of time covered by each load
        "r8s_std": 1,
    },
    "SCHEDULE": {
        "r8s_period_days": 14,
        "r8s_cpu_load": 50,
        "r8s_memory_load": 50,
        "r8s_avg_disk_iops": -1,
        "r8s_max_disk_iops": -1,
        "r8s_net_output_load": -1,
        "r8s_std": 1,
        "r8s_cron_start": '0 9 * * 1-5',
        "r8s_cron_stop": '0 18 * * 1-5',
    },
}
```

To simplify the testing process, test case examples are prepared and presented below based on the predefined dataset:

- To verify getting several recommended actions in the result the following tags should be set:

```
{
  "r8s_test_case": "schedule", // for tags that are not specified, we want
to use default values for       SCHEDULE
  "r8s_cpu_load": 90, // overwriting cpu_load to force SCALE_UP
  "r8s_memory_load": 85, // overwriting memory_load to force SCALE_UP
}
```

The result will be ['SCHEDULE', 'SCALE_UP'].

- To verify detecting a specific schedule on a specific time scale the following tags should be set:

```
{
  "r8s_test_case": "SCHEDULE", // for tags that are not specified, we want
to use default values for SCHEDULE

  "r8s_period_days": 60, // to generate 60 days of metrics

  "r8s_cron_start": '0 12 * * 2-6',

  "r8s_cron_stop": '0 14 * * 2-6', // Instance is working from 12:00 to
14:00, Tuesday to Saturday included

}
```

- To verify detecting a split load that contains of 4 different load types the following tags should be set:

```
{

  "r8s_test_case": "SPLIT", // for tags that are not specified, we want to
use default values for SPLIT

  "r8s_cpu_load": [25, 50, 70, 95],

  "r8s_memory_load": [25, 50, 70, 95],

  "r8s_probability": [25, 25, 25, 25], // means that each of load type
(25%/50%/75%/95%) covers 25% of time

}
```

Note:

- The r8s_probability values may not add up to 100. For example, [10, 5, 5] probability will end up with a 50%/25%/25% probability.
- If overwriting default values for SPLIT case, make sure that number of periods matches for all CPU/memory/probability. CPU/memory loads must be within 0-100.

4 PRODUCT INTEGRATION

The section below provides the step by step instruction on RightSizer integration with third-party systems

4.1 PREREQUISITES

Before integrating with the RightSizer, you need to have the following:

- r8s api host
- r8s user

4.2 INTEGRATION STEPS

To integrate RightSizer with a third-party solution, do the following:

1. Create a Storage (storing reports):
   a. Using r8s CLI:

```
r8s storage s3 add --storage_name maestro_rightsizer_storage --type STORAGE
--bucket_name $BUCKET_NAME --prefix $PREFIX
```

   b. Using r8s API:

```
URL: $HOST_URL/storages
HttpMethod: POST
BODY:
{
  "name": "$STORAGE_NAME",
  "service": "S3_BUCKET",
  "type": "STORAGE",
  "access": {
    "bucket_name": "$BUCKET_NAME",
    "prefix": "$PREFIX"
  }
}
```

2. Create Data Source: (metrics storage)
   a. Using r8s CLI:

```
r8s storage s3 add --storage_name $DATA_SOURCE_NAME --type DATA_SOURCE --
bucket_name $BUCKET_NAME --prefix $PREFIX
```

   b. Using r8s API:

```
URL: $HOST_URL/storages
HttpMethod: POST
BODY:
{
  "name": "$STORAGE_NAME",
  "service": "S3_BUCKET",
  "type": "DATA_SOURCE",
  "access": {
    "bucket_name": "$BUCKET_NAME",
    "prefix": "$PREFIX"
  }
}
```

3. Create Job Definition
    a. Using r8s cli:

```
r8s definition add --definition_name $DEFINITION_NAME --customer $CUSTOMER -
-data_source_name $DATA_SOURCE --storage_name $STORAGE --model_name $MODEL
```

    b. Using r8s API:

```
URL: $HOST_URL/jobs/definitions
HttpMethod: POST
BODY:
{
  "name": "$DEFINITION_NAME",
  "customer": "$CUSTOMER",
  "data_source": "$DATA_SOURCE",
  "storage": "$STORAGE",
  "model": "$MODEL"
}
```

4. Upload instances metric files to $DATA_SOURCE destination
5. Submit a job:
    a. Using r8s cli:

```
r8s job submit --definition_name maestro_job_definition --scan_timestamp
$TIMESTAMP --scan_customer $CUSTOMER --scan_tenants $TENANT_1 --scan_tenants
$TENANT_2
```

    b. Using r8s API:

```
URL: $HOST_URL/jobs/
HttpMethod: POST
BODY:
{
  "job_definition": "$DEFINITION_NAME",
  "customer": "$CUSTOMER",
  "tenants": ["$TENANT1", "$TENANT2"],
  "scan_timestamp": "$$TIMESTAMP"
}
```

6. Describe the job:
    a. Using r8s cli:

```
r8s job describe --job_id $JOB_ID
```

b. Using r8s API:

```
URL: $HOST_URL/jobs/
HttpMethod: GET
BODY:

{

  "id": "$JOB_ID"

}
```

7. Download report (instance recommendations)
   a. Using r8s cli

```
r8s report download --job_id $JOB_ID
```

b. Using r8s API:

```
URL: $HOST_URL/reports/
HttpMethod: GET
BODY:



{

   "report_type": "download",

   "id": "$JOB_ID",

   "customer": "$CUSTOMER",

   "tenant": "$TENANT",

   "region": "$REGION",

}
```

5 TABLE OF FIGURES

6 VERSION HISTORY

Version	Date	Summary
1.0	May 29, 2024	- Initial version created