Validation Project execution

# Use Case #6

Junior

* Provision of storage services

Middle

* Provision and setup serverless infrastructure
* Write a script
* Custom metrics for monitoring and logging

Senior

* Add logging to the script
* Write a script, using function(s), object(s) (if possible), class(es) (if possible)
* Configure access to the objects in storage services

## Use Case Title

Provision and set up serverless infrastructure in cloud environments

## Use Case Description

The objective of this use case is to design and develop a serverless application that collects specific infrastructure metrics.

## Participant information

|  |  |
| --- | --- |
| **Item** | **Description** |
| Name | Marcin Winnik |
| Email | mwinn@softserveinc.com |
| Job Profile + level (ex. Node.js Middle, Java Senior) | Senior AWS DevOps Engineer |
| PeEX Primary Profile |  |
| PeEX Profile Coverage | Overall = 62%  L4 = 22,63%  L3 = 67,44%  L2 = 87,96%  L1 = 100% |
| Date |  |
| Generative AI used (none, ChatGPT 3.5, GPT-4, StarCoder, etc) | None |

## Experience / Maturity levels definition\*

In some sections of this document, you will be asked to provide self-evaluation mark regarding your experience or maturity with some area, topic, tool, methodology. Use the definition below to provide your mark.

**None**: Not familiar with the subject matter and has no prior experience or knowledge in the area, tool, methodology

**Beginner**: Basic familiarity with the tool, methodology or concept, knowing about it and occasionally using it for simple use cases, tasks

**Competent**: Regularly uses the tool, methodology or concept in everyday duties and have a good level of proficiency and understanding.

**Proficient**: Advanced experience and knowledge, utilizing the tool, methodology or concept in advanced activities, understanding its intricacies, and being able to educate others about it.

## Use Case self-assessment\*

|  |  |
| --- | --- |
| **Area** | **Experience / Maturity**\* |
| What is your experience with Use Case? | Competent |

## Tools and Methodologies

To keep consistency Allowed to use only specified tool and methodologies during development.

**All technologies**

|  |  |
| --- | --- |
| **Tool / Methodology** | **Experience / Maturity**\* |
| GitHub | Competent |
| Python | Competent |

**AWS**

|  |  |
| --- | --- |
| **Tool / Methodology** | **Experience / Maturity**\* |
| AWS Lambda | Competent |
| AWS EventBridge | Competent |
| Amazon S3 | Proficient |
| AWS SDK for Python | Beginner |

## Acceptance criteria:

* The function is created by using Python.
* The function has a role with permissions to collect metrics and write them to a storage.
* The function is tested and verified to collect the required metrics once a day using cloud SDK (for scheduling in AWS use AWS EventBridge, for scheduling in GCP use Cloud Scheduler which calls HTTP triggered Cloud Function).
* The function stores the collected metrics as JSON files in cloud storage.
* The storage is configured with encryption, lifecycle policies, and access control.

## Quality Details:

* All commented code is removed from the project.
* Proper exception handling is incorporated in the code.
* Passwords or other secrets are not visible as plain text in the code.
* Sensitive logs are appropriately suppressed.
* Hardcoding of values is avoided.

## Use Case Implementation

### Environment Preparation

* Create new chat conversation in AI system to discuss the implementation details and clarify any questions or concerns.
* Setup cloud infrastructure by creating 1 unattached disk, 2 non-encrypted disks, 3 non-encrypted snapshots.
* Create a public repository on GitHub or use provided repo for you. Make sure to make your first commit UC#6 with changes to Readme.md, as time of this commit will be considered as project start. Add commit message “UC#6 Start”.
* Setup serverless service and storage and in the cloud.
* Install the necessary packages for function development into the newly created venv using a requirements file.

|  |  |
| --- | --- |
| **Repository** | **Link** |
| Provide Link to created GitHub repo |  |

Application Creation

* Write Python code that collects the metrics from the number and overall size of unattached disk volumes, not encrypted disk volumes, and not encrypted snapshots.
* Deploy the code to the serverless service.
* Configure a schedule to run the serverless service once a day.
* Configure permissions to collect metrics and to store them in a storage service.
* Verify if the collected metrics are stored in the storage and storage is encrypted, lifecycle policies, and access control are configured.
* Publish the code to the source code management system.
* Store the AI chat conversation in file “Application Creation.txt”. If you did the work without AI support, create an empty file.
* Commit the code and relevant configuration files to the source code management (SCM) repository with message “UC#6 Application Creation”.
* The quality of the files will be validated manually.

Note: Throughout the implementation, ensure adherence to the quality details mentioned above, such as removing commented code, incorporating exception handling, securing passwords and secrets, suppressing sensitive logs, and avoiding hardcoded values.

**Estimated Time**: 4 hours.

Participant feedback

|  |  |
| --- | --- |
| **Item** | **Description** |
|  |  |
|  |  |
|  |  |
|  |  |

Expert feedback

|  |  |  |
| --- | --- | --- |
| **Criterion** | **Score (1-4)\*** | **Notes** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**\*4 - excellent (smth was done with extra mile);**

**3 - good (everything was done according to acceptance criteria);**

**2 – bad (has gaps, need some extra time for completion);**

**1 - the worst (a lot of mistakes/no result/step was omitted, must be disqualified).**

|  |
| --- |
| **Comment** |
|  |

|  |  |
| --- | --- |
| **Final Score** | **Evaluation\*** |
| Implementation score (solution suitability in accordance with the functional criteria) |  |
| Quality score (quality of the implemented solution) |  |