AWS Compute Blog

AWS Compute Optimizer supports AWS Graviton migration guidance

by Pranaya Anshu | on 10 JAN 2022 | in AWS Compute Optimizer, Compute, Graviton | Permalink | Share

This post is written by Letian Feng, Principal Product Manager for AWS Compute Optimizer, and Steve Cole, Senior EC2 Spot Specialist Solutions Architect.

Today, AWS Compute Optimizer is launching a new capability that makes it easier for you to optimize your EC2 instances by leveraging multiple CPU architectures, including x86-based and AWS Graviton-based instances. Compute Optimizer is an opt-in service that recommends optimal AWS resources for your workloads to reduce costs and improve performance by analyzing historical utilization metrics. AWS Graviton processors are custom-built by Amazon Web Services using 64-bit Arm cores to deliver the best price performance for your cloud workloads running in Amazon EC2, with the potential to realize up to 40% better price performance over comparable current generation x86-based instances. As a result, customers interested in Graviton have been asking for a scalable way to understand which EC2 instances they should prioritize in their Graviton migration journey. Starting today, you can use Compute Optimizer to find the workloads that will deliver the biggest return for the smallest migration effort.

How it works

Compute Optimizer helps you find the workloads with the biggest return for the smallest migration effort by providing a migration effort rating. The migration effort rating, ranging from very low to high, reflects the level of effort that might be required to migrate from the current instance type to the recommended instance type, based on the differences in instance architecture and whether the workloads are compatible with the recommended instance type.

Clues about the type of workload running are useful for estimating the migration effort to Graviton. For some workloads, transitioning to Graviton is as simple as updating the instance types and associated Amazon Machine Images (AMIs) directly or in various launch or CloudFormation templates. For other workloads, you might need to use different software versions or change source codes. The quickest and easiest workloads to transition are Linux-based open-source applications. Many open source projects already support Arm64, and by extension Graviton. Therefore, many customers start their Graviton migration journey by checking whether their workloads are among the list of Graviton-compatible applications. They then combine this information with estimated savings from Compute Optimizer to build a list of Graviton migration opportunities.

Because Compute Optimizer cannot see into an instance, it looks to instance attributes for clues about the workload type running on the EC2 instance. The clues Compute Optimizer uses are based on the instance attributes customers provide, such as instance tags, AWS Marketplace product names, AMI names, and CloudFormation templates names. For example, when an instance is tagged with “key:application-type” and “value:hadoop”, Compute Optimizer will identify the application –Apache Hadoop in this example. Then, because we know that major frameworks, such as Apache Hadoop, Apache Spark, and many others, run on Graviton, Compute Optimizer will indicate that there is low migration effort to Graviton, and point customers to documentation that outlines the required steps for migrating a Hadoop application to Graviton.

As another example, when Compute Optimizer sees an instance is using a Microsoft Windows SQL Server AMI, Compute Optimizer will infer that SQL Server is running. Then, because it takes a lot of effort to modernize and migrate a SQL Server workload to Arm, Compute Optimizer will indicate that there is a high migration effort to Graviton. The most effective way to give Compute Optimizer clues about what application is running is by putting an "application-type" tag onto each instance. If Compute Optimizer doesn't have enough clues, it will indicate that it doesn't have enough information to offer migration guidance.

The following shows the different levels of migration effort:

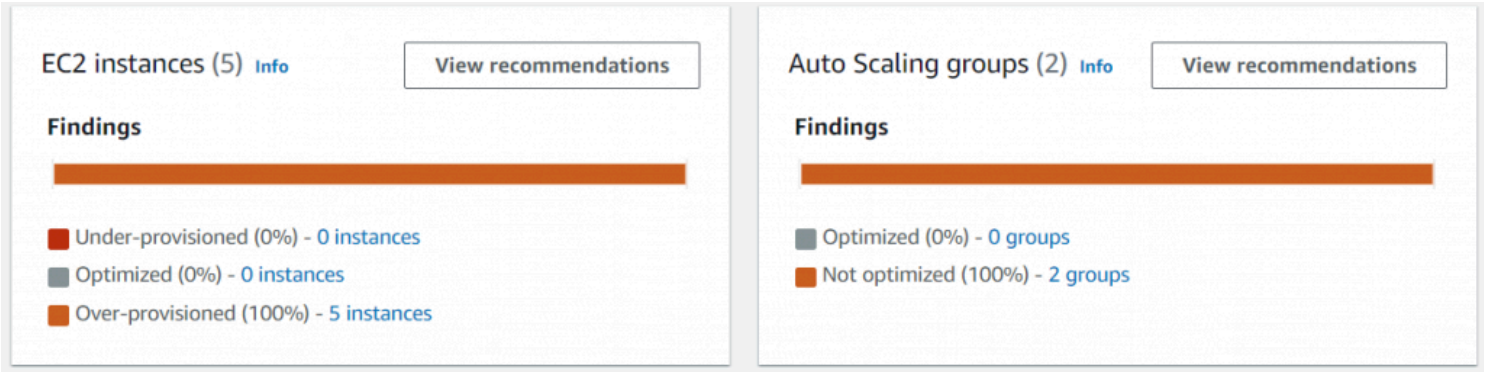
- **Very Low** – The recommended instance type has the same CPU architecture as the current instance type. Often, customers can just modify instance types directly, or do a simple re-deployment onto the new instance type. So, this is just an optimization, not a migration.
- **Low** – The recommended instance type has different CPU architecture from the current instance type, but there's a low-effort migration path. For example, migrating Apache Hadoop or Redis from x86 to Graviton falls under this category as both Hadoop and Redis have Graviton-compatible versions.
- **Medium** – The recommended instance type has different CPU architecture from the current instance type, but Compute Optimizer doesn't have enough information to offer migration guidance.
- **High** – The recommended instance type has different CPU architecture from the current instance type, and the workload has no known compatible version on the recommended CPU architecture. Therefore, customers may need to re-compile their applications or re-platform their workloads (like moving from SQL Server to MySQL).

More and more applications support Graviton every day. If you're running an application that you know has low migration effort, but Compute Optimizer isn't yet aware, please tell us! Shoot us an email at compute-optimizer-feedback@amazon.com with the application type, and we'll update our migration guidance mappings as quickly as we can. You can also put an "application-type" tag on your instances so that Compute Optimizer can infer your application type with high confidence.

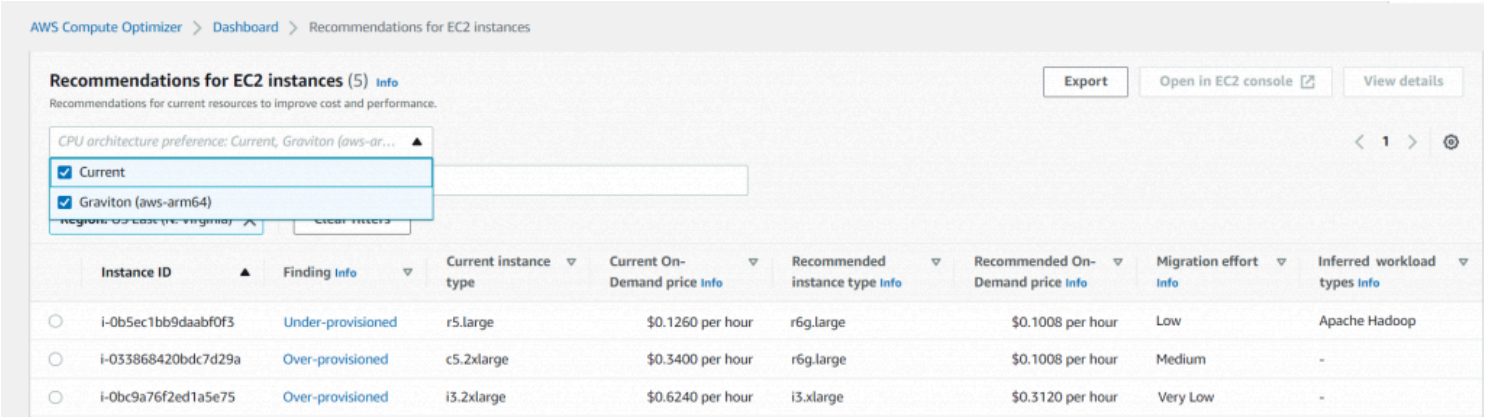
Customers who have already opted into Compute Optimizer recommendations will have immediate access to this new capability. Customers who haven't can opt-in with a single console click or API, enabling all Compute Optimizer features.

Walk through

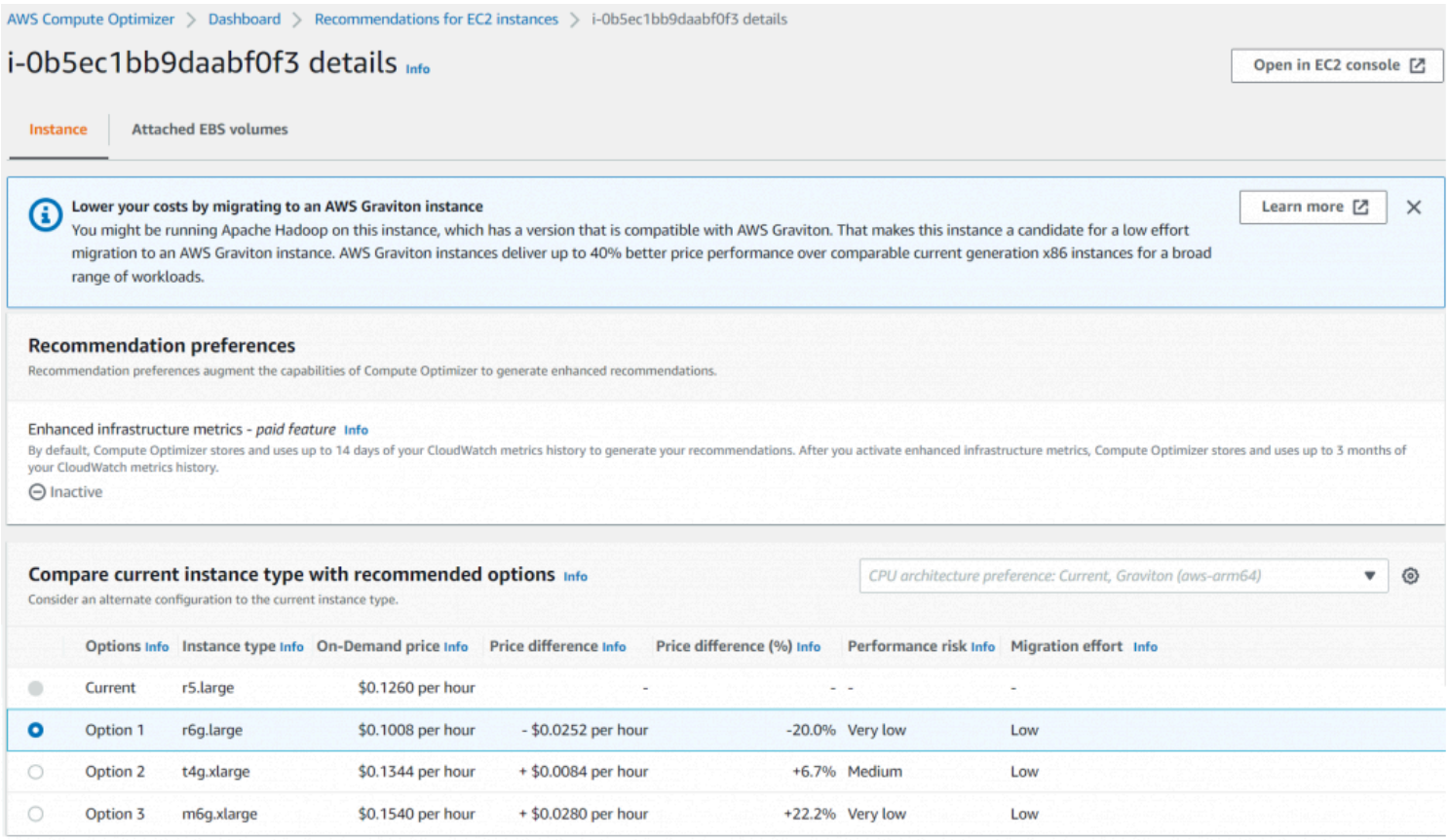
Now, let's take a look at how to get started with Graviton recommendation on Compute Optimizer. When you open the Compute Optimizer console, you will see the dashboard page that provides you with a summary of all optimization opportunities in your account. Graviton recommendation is available for EC2 instances and Auto Scaling groups.



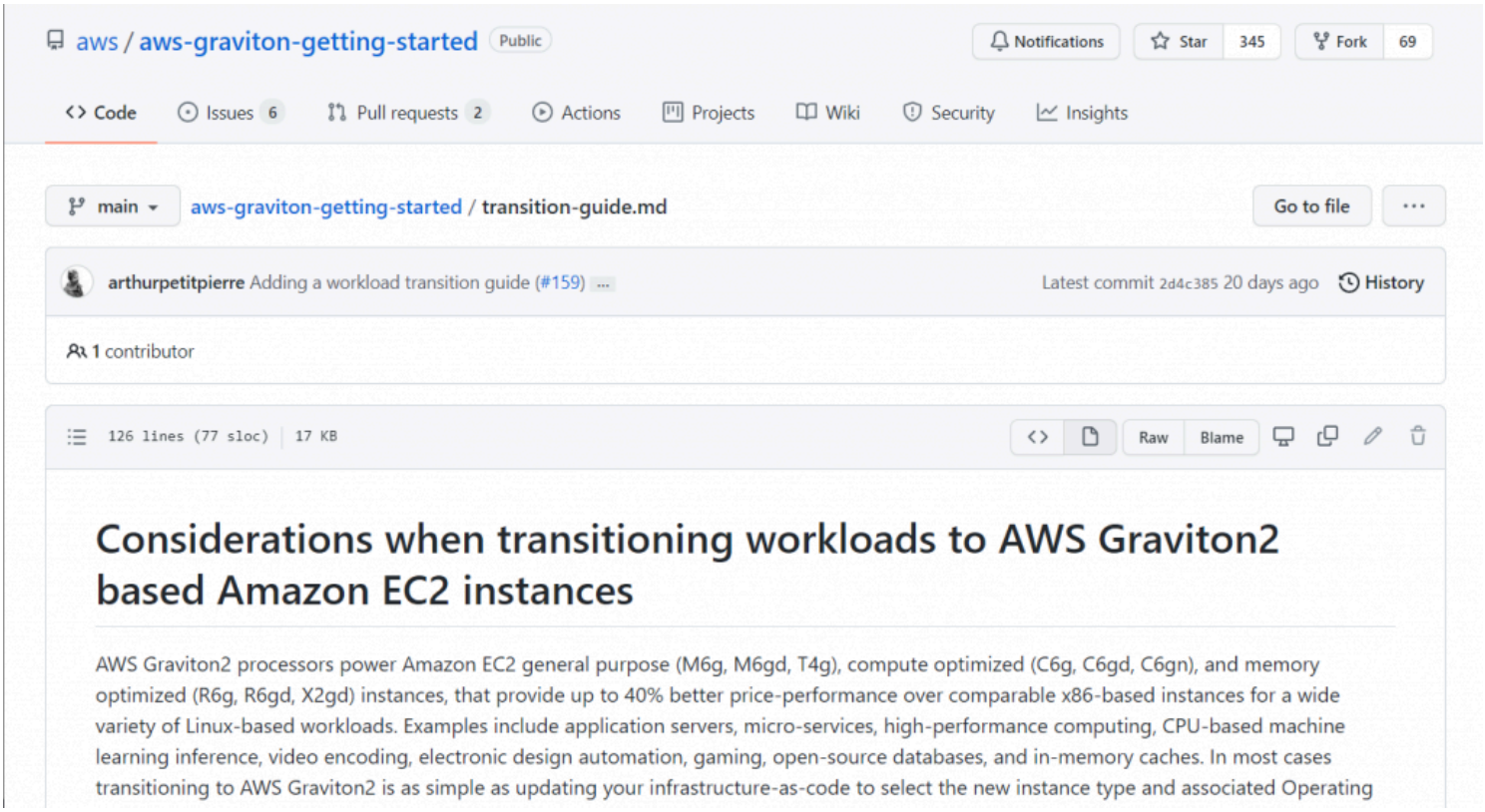
After you click on **View recommendations for EC2 instances**, you will come to the EC2 recommendation list view. Here is where you can see a list of your EC2 instances, their current instance type, our finding (over-provisioned, under-provisioned, or optimized), the recommended optimal instance type, and the estimated savings if there is a downsizing opportunity. By default, we will show you the best-fit instance type for the price regardless of CPU architecture. In many cases this means that Graviton will be recommended because EC2 offers a wide selection of Graviton instances with comparatively high price/performance ratio. If you'd like to only look at recommendations with your current architecture, you can use the **CPU architecture preference** dropdown to tell Compute Optimizer to show recommendations with only the current CPU architecture.



Here you can see two new columns — **Migration effort** and **Inferred workload types**. The **Inferred workload types** field shows the type of workload Compute Optimizer has inferred your instance is running. The **Migration effort** field shows how much effort you might need to spend if you migrate from your current instance type to recommended instance type based on the inferred workload type. When there is no change in CPU architecture (i.e. moving from an x86-instance type to another x86-instance type, like in the third row), the migration effort will be **Very low**. For x86-instances that are running Graviton-compatible applications, such as Apache Hadoop, NGINX, Memcached, etc., when you migrate the instance to Graviton, the effort will be **Low**. If Compute Optimizer cannot identify the applications, the migration effort from x86 to Graviton will be **Medium**, and you can provide application type data by putting an application-type tag key onto the instance. You can click on each row to see more detailed recommendation. Let's click on the first row.



Compute Optimizer identifies this instance to be running Apache Hadoop workloads because there's Amazon EMR system tag associated with it. It shows a banner that details why Compute Optimizer considers this as a low-effort Graviton migration candidate, and offers a migration guide when you click on **Learn more**.



The same Graviton recommendation can also be retrieved through Compute Optimizer API or CLI. Here's a sample CLI that retrieves the same recommendation as discussed above:

```
aws compute-optimizer get-ec2-instance-recommendations --instance-arns arn:aws:ec2:us-
{
    "instanceRecommendations": [
        {
            "instanceArn": "arn:aws:ec2:us-west-2:000000000000:instance/i-0b5ec1bb9daa
            "accountId": "000000000000",
            "instanceName": "Compute Intensive",
            "currentInstanceType": "r5.large",
            "finding": "UNDER_PROVISIONED",
            "findingReasonCodes": [
                "CPUUnderprovisioned",
                "EBSIOPSOverprovisioned"
            ],
            "inferredWorkloadTypes": [
                "ApacheHadoop"
            ],
            "utilizationMetrics": [
                {
                    "name": "CPU",
                    "statistic": "MAXIMUM",
                    "value": 100.0
                },
                {
                    "name": "EBS_READ_OPS_PER_SECOND",
                    "statistic": "MAXIMUM",
                    "value": 0.0
                },
                {
                    "name": "EBS_WRITE_OPS_PER_SECOND",
                    "statistic": "MAXIMUM",
                    "value": 4.943333333333333
                },
                {
                    "name": "EBS_READ_BYTES_PER_SECOND",
                    "statistic": "MAXIMUM",
                    "value": 0.0
                },
                {
                    "name": "EBS_WRITE_BYTES_PER_SECOND",
                    "statistic": "MAXIMUM",
```

```
                "value": 880541.9921875
            },
            {
                "name": "NETWORK_IN_BYTES_PER_SECOND",
                "statistic": "MAXIMUM",
                "value": 18113.96638888889
            },
            {
                "name": "NETWORK_OUT_BYTES_PER_SECOND",
                "statistic": "MAXIMUM",
                "value": 90.37638888888888
            },
            {
                "name": "NETWORK_PACKETS_IN_PER_SECOND",
                "statistic": "MAXIMUM",
                "value": 2.484055555555556
            },
            {
                "name": "NETWORK_PACKETS_OUT_PER_SECOND",
                "statistic": "MAXIMUM",
                "value": 0.3302777777777778
            }
        ],
        "lookBackPeriodInDays": 14.0,
        "recommendationOptions": [
            {
                "instanceType": "r6g.large",
                "projectedUtilizationMetrics": [
                    {
                        "name": "CPU",
                        "statistic": "MAXIMUM",
                        "value": 70.76923076923076
                    }
                ],
                "platformDifferences": [
                    "Architecture"
                ],
                "migrationEffort": "Low",
                "performanceRisk": 1.0,
                "rank": 1
            },
            {
                "instanceType": "t4g.xlarge",
```

```
            "projectedUtilizationMetrics": [
                {
                    "name": "CPU",
                    "statistic": "MAXIMUM",
                    "value": 33.33333333333333
                }
            ],
            "platformDifferences": [
                "Hypervisor",
                "Architecture"
            ],
            "migrationEffort": "Low",
            "performanceRisk": 3.0,
            "rank": 2
        },
        {
            "instanceType": "m6g.xlarge",
            "projectedUtilizationMetrics": [
                {
                    "name": "CPU",
                    "statistic": "MAXIMUM",
                    "value": 33.33333333333333
                }
            ],
            "platformDifferences": [
                "Architecture"
            ],
            "migrationEffort": "Low",
            "performanceRisk": 1.0,
            "rank": 3
        }
    ],
    "recommendationSources": [
        {
            "recommendationSourceArn": "arn:aws:ec2:us-west-2:000000000000:ins
            "recommendationSourceType": "Ec2Instance"
        }
    ],
    "lastRefreshTimestamp": "2021-12-28T11:00:03.576000-08:00",
    "currentPerformanceRisk": "High",
    "effectiveRecommendationPreferences": {
        "cpuVendorArchitectures": [
            "CURRENT",
```

```
                    "AWS_ARM64"
                ],
                "enhancedInfrastructureMetrics": "Inactive"
            }
        }
    ],
    "errors": []
}
```

Conclusion

Compute Optimizer Graviton recommendations are available in in US East (Ohio), US East (N. Virginia), US West (N. California), US West (Oregon), Asia Pacific (Mumbai), Asia Pacific (Seoul), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), Canada (Central), Europe (Frankfurt), Europe (Ireland), Europe (London), Europe (Paris), Europe (Stockholm), and South America (São Paulo) Regions at no additional charge. To get started with Compute Optimizer, visit the Compute Optimizer webpage.

TAGS: AWS Compute Optimizer, AWS Graviton