ARM Application Deployment on EKS

Downloaded from Epic Games Confluence
Date: 2025-07-12 04:09:06
Original URL: https://confluence-epicgames.atlassian.net/wiki/spaces/CDE/pages/81068721

This page provides instructions for workload developers/owners to enable their workload deployment on ARM64 Graviton nodes.

Assumptions and pre-requisites

- Your application container has an ARM64 build available for it
- If you are using epic-app, be sure that your application imports epic-app version `3.1.1` or greater in your `Chart.yaml` file. This is the first epic-app version to support arm enablement.

- Your application does not use a https://confluence-epicgames.atlassian.net/wiki/spaces/CE/pages/93488602 sidecar for authentication (UAP does not currently have an ARM image)

⚠️ Before you decide to move your workload to ARM-based Graviton nodes, it is highly recommended to do some performance benchmarking to ensure your workload performs well on ARM instance types. The HLD (link provided under Reference section) has a *Performance benchmarking* section that provides information on benchmarking results for many standard architectures. A **true** picture of a workloads' performance on AWS Graviton nodes can only be assessed by doing performance benchmarking of your unique workload. That said, it is very unusual to see any loss in performance for applications of any type.

Deploying ARM application with epic-app chart as base

To enable ARM deployment, all you need to do is add a single enable setting in your Helm chart's `values.yaml` file and re-deploy your app:

values.yaml

```
arm:
  enabled: true
```

Setting of this flag to `false` or absence of this attribute all together, defaults to standard x86 arch deployment for backward compatibility. Enabling this flag by setting it to `true` will set your application to deploy your workload on ARM64 Graviton nodes. As your application executes a rolling deployment, newly created pods will be provisioned onto ARM64 nodes. If these pods are unable to start properly or pass health checks, your deployment will halt as would any Kubernetes deployment with unhealthy pods coming online. To roll back your application off of ARM nodes, simply set `enabled` to `false` or delete the `arm` section entirely and re-deploy.

Deploying ARM applications without epic-app chart as base

In the case that you do not use `epic-app`, this is how you deploy on ARM nodes. Augment your podSpec with affinity & tolerations specific to enabling ARM deployment, as shown below:

Affinity & tolerations

```
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
          - matchExpressions:
            - key: "kubernetes.io/arch"
              operator: "In"
              values:
              - arm64
  tolerations:
    - key: "epicgames.com/arm"
      effect: "NoExecute"
      operator: "Exists"
    - key: "epicgames.com/arm"
      effect: "NoSchedule"
      operator: "Exists"
```

In case you have existing affinity and/or tolerations in your podSpec, you will need to combine yours with what's shown above.

ARM64 builds

AWS Guidance

AWS maintains optimization guides for common programming languages, which are good to check before switching your application to ARM:

https://github.com/aws/aws-graviton-getting-started/

How to find out whether a container supports ARM64

You can `docker inspect` your image's manifest in the registry to find out whether your image supports arm64. Command shown below:

inspect manifest example

```
$ docker manifest inspect <registry url>

{
   "schemaVersion": 2,
   "mediaType": "application/vnd.docker.distribution.manifest.list.v2+j
   "manifests": [
      {
         "mediaType": "application/vnd.docker.distribution.manifest.v2+
         "size": 738,
         "digest": "sha256:8e89f0ddbd0f8586b3803ca1084e5b2cb562f874dd99
         "platform": {
            "architecture": "amd64",
            "os": "linux"
         }
      },
      {
         "mediaType": "application/vnd.docker.distribution.manifest.v2+
         "size": 950,
         "digest": "sha256:a8f2def6bf480f8dee521fd7422109b19c78260e81c6
```

```
            "platform": {
               "architecture": "amd64",
               "os": "windows",
               "os.version": "10.0.17763.3770"
            }
         },
         {
            "mediaType": "application/vnd.docker.distribution.manifest.v2+
            "size": 738,
            "digest": "sha256:6431c2212d19ea3b333a754f8c4ddfb2b71db7ebb8cb
            "platform": {
               "architecture": "arm",
               "os": "linux",
               "variant": "v7"
            }
         },
         {
            "mediaType": "application/vnd.docker.distribution.manifest.v2+
            "size": 738,
            "digest": "sha256:5cba3f08f51b10082cdc19538b0201f90b51cb9f3ea9
            "platform": {
               "architecture": "arm64",
               "os": "linux"
            }
         }
      ]
}
```


Look for `architecture: arm64` in the output.

Building ARM Images

Manually building ARM64 builds

Refer to this link for step-by-step instructions to manually build multi-architecture images: https://www.docker.com/blog/getting-started-with-docker-for-arm-on-linux/

Building an ARM64 container with Github Actions

Refer to this link for instructions to build multi-architecture images with Github actions: https://docs.docker.com/build/ci/github-actions/multi-platform/

Example implementation: https://github.ol.epicgames.net/actions/docker_setup-buildx-action/blob/master/.github/workflows/ci.yml

Building an ARM64 container with Codefresh

Refer to this link for building multi-architecture images in Codefresh pipelines: https://confluence-epicgames.atlassian.net/wiki/spaces/CE/pages/93490515

Building an ARM64 container with TeamCity

Refer to his Confluence page for instructions to setup a TeamCity job for running multi-architecture builds: https://confluence-epicgames.atlassian.net/wiki/spaces/CE/pages/93489201

Example implementation: https://teamcity.ol.epicgames.net/project/MultiarchBuildTest?mode=builds

---

Page Information:
Page ID: 81068721
Space: Cloud Developer Platform
Downloaded: 2025-07-12 04:09:06