

# Multi-Arch SIG Report

The Multi-Arch SIG was formed in December 2019, so it has been a year since then. We generated this report to provide an update on what happened with multi-arch in the OpenStack community within the past year.

## What is the Multi-arch SIG?

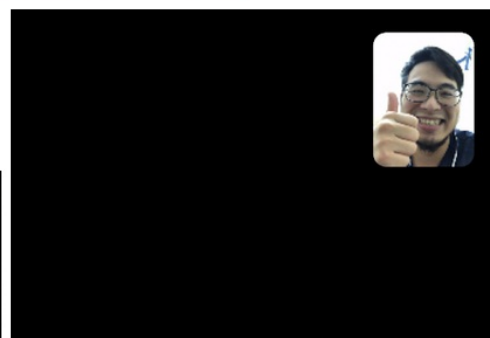
In short, the Multi-Arch SIG is targeting all of the below work so that OpenStack can have better integration with different CPU architectures. We had a lot of discussions when this SIG was just an idea during the Shanghai Summit (Nov. 2019). Discussions started from supporting ARM64 only and soon adopting ideas to support other CPU architectures as we found most CPU architecture support status are facing the same problems and require very similar changes (in a lot of cases, it requires the same set of changes). From there, the SIG appears to be the best place to gather experiences from one CPU architecture support to another. Those experiences can be documented and provided to the entire OpenStack community and to users who might be interested. So the Multi-Arch SIG was founded to support such goals. Remember, the SIG is a place for collaboration and extends across affiliations.

## Join the Multi-Arch SIG!

We hope to help the community to reach better support for multi-arch, therefore we hope to have more help, more hands, and more resources to achieve that goal.

Please join us in the coming [PTG](#) (Our [schedule and etherpad](#)).

We have a [SIG etherpad](#) that launched on day one of the Multi-Arch SIG and will keep updated information and links on it. Also, you can go check on the [Etherpad](#) from the meeting of the Multi-Arch SIG at the VPTG in 2020. Those two etherpads should give you good initial information about this SIG and what this SIG is working on.



We will also try to compile community efforts (from this SIG or not) on the SIG's StoryBoard. If you have any questions, ideas, or information about ongoing work regarding multiple-architecture support for OpenStack, feel free to reach out. Here are some places where you can find us:

- IRC: #openstack-multi-arch
- Meeting (on demand): [http://eavesdrop.openstack.org/#Multi-Arch\\_SIG\\_Meeting](http://eavesdrop.openstack.org/#Multi-Arch_SIG_Meeting)
- ML: use "[Multi-arch SIG]" or "[Multi-arch]" tags on Mailing list [OpenStack Discuss](#)
- [StoryBoard](#): We use this StoryBoard to collect any multi-arch efforts in the community. Also to collect user stories or bugs too.
- If any part fails, or you find any difficulty, join us. Feel free to contact the SIG chairs or any SIG member to provide services/consult or onboarding you. Current [SIG chairs](#): [Rico Lin](#) (irc: ricolin)

Let us know what your interests are or your story about OpenStack multi-arch support is. We will try to continue generating reports (hopefully with your story or work shared inside) once more progress is made. So please stay tuned. : )

For those who don't know what multi-arch means...

## What is multi-arch?

Multi-arch is a convenient name for the effort to have OpenStack work on any (desired) architecture. There are many distinct areas of focus within this effort.

## CI and testing

We need to test OpenStack on multiple architectures. Ideally, there should be upstream CI for any relevant architecture. New test cases can also be considered.

## Packaging and containers and images

Packages and containers should be built for all architectures. Keep in mind the Guidelines for Managing Releases of Binary Artifacts.

## Deployment tools

OpenStack deployment tools should support deploying on non-x86 architecture. In other words, tools should support any special steps needed for non-x86. Support for deploying a cloud with a mix of architectures may also be valuable. Note that deployment tools are typically opinionated about what content they deploy, so multi-arch support in deployment tools may be dependent upon the availability of artifacts.

## Virt drivers

Running OpenStack often means virtual machines managed by Nova. From a multi-arch perspective, this means awareness of Virt drivers. More specifically, the Libvirt KVM driver should work on non-x86 architectures (it already does) and the accuracy of the Feature Support Matrix should be maintained. There are also architecture-specific virtualization technologies such as PowerVM and z/VM with Virt drivers in Nova that are part of the multi-arch landscape.

## Evangelism

The value of multi-arch should be made clear.

## Why does support for multi-arch in OpenStack matter?

Testing of OpenStack is a great sandbox for proving capabilities of CPU architectures because OpenStack touches so many pieces of a linux system. It interacts with networking, virtualization, filesystems, disk management, python, web servers, database servers, amqp servers the list goes on (and on). If you can run OpenStack there is a good chance you can run most other software.

Also, x86 is seeing viable alternatives that may offer better performance per dollar, or better performance per watt, in some cases.

## Where are we for multi-arch support?

Testing of OpenStack is a great sandbox for proving capabilities of CPU architectures because OpenStack touches so many pieces of a linux system. It interacts with networking, virtualization, filesystems, disk management, python, web servers, database servers, amqp servers the list goes on (and on). If you can run OpenStack there is a good chance you can run most other software.

Also, x86 is seeing viable alternatives that may offer better performance per dollar, or better performance per watt, in some cases.

## Where are we for multi-arch support?

Currently, the x86 CPU architecture is still the only one that has been fully tested by the OpenStack community; however, there are already cases showing OpenStack can be fully deployed on multiple CPU architectures in production. There are already vendors who deliver OpenStack with ARM64 nodes for production. Linaro even generously donated an ARM64 OpenStack [environment](#) to OpenInfra Foundation (Thanks to Marcin Juszkiewicz, Kevin Zhao, and the Infra team who made this happen). This is currently the only non-x86 CI environment we have in our community. For more information about this Linaro environment, you can check out the superuser article [`How Arm is becoming a first class citizen in OpenStack`](#).

## More good news!

Regarding Arm64 CI resources, [Oregon State University Open Source Lab](#) (OSUOSL) is also generously donating about 15+ arm64 nodes to the community. And the environment is now ready for OpenStack community to use ([system config is merged](#)).

## Related work in the OpenStack community

### ARM64 unit test coverage

We currently have a [unit test job](#) (work by Rico Lin) for testing on the ARM64 CPU architecture.

The zuul job template ``openstack-python3-wallaby-jobs-arm64`` currently only contains a zuul job ``openstack-tox-py38-arm64``. The job will run unit tests in a python 3.8 environment on arm64 architecture.

```

616 - project-template:
617     name: openstack-python3-wallaby-jobs-arm64
618     description: |
619         Runs unit tests for an OpenStack Python project under the CPython
620         version 3 releases designated for testing in the Wallaby release
621         on top of ARM64 architecture. This currently runs only a limited
622         test set and supplements the python3-wallaby-jobs. And it should
623         not be run instead of the default!
624     check-arm64:
625         jobs:
626         - openstack-tox-py38-arm64:
627             voting: false
  
```

It's currently proposed as a non-voting job on a separate pipeline "check-arm64" which helps teams adopt this job as it does not affect current development and review workflow for project teams, plus offers a chance to check how patches work on ARM64 environments. This job is currently running and tested on [some core projects](#) (like Nova, Neutron, Ironic, Keystone, etc.). Once all core projects (and other projects which might require testing against multiple CPU architectures) adopt this job, we can start to provide a better guarantee for running OpenStack services on ARM64. Also, it's possible now to take advantage of Zuul and check the [status/health](#) of the job.

Job	Project	Bran...	Pipeline	Change	Duration	Start time	Result
✘ openstack-tox-py38-arm64 (non-voting)	openstack/neutron	master	check-arm64	<a href="#">776567,2</a>	47 mins 29 secs	2021-02-19 05:03:46	FAILURE
✔ openstack-tox-py38-arm64 (non-voting)	openstack/neutron	master	check-arm64	<a href="#">776568,2</a>	48 mins 3 secs	2021-02-19 05:05:06	SUCCESS
✔ openstack-tox-py38-arm64 (non-voting)	openstack/nova	master	check-arm64	<a href="#">775308,3</a>	27 mins 51 secs	2021-02-19 04:48:15	SUCCESS
✔ openstack-tox-py38-arm64 (non-voting)	openstack/neutron	master	check-arm64	<a href="#">773283,5</a>	1 hr 0 mins 48 secs	2021-02-19 02:07:49	SUCCESS
✔ openstack-tox-py38-arm64 (non-voting)	openstack/neutron	master	check-arm64	<a href="#">773281,4</a>	58 mins 28 secs	2021-02-19 02:07:21	SUCCESS
✔ openstack-tox-py38-arm64 (non-voting)	openstack/neutron	master	check-arm64	<a href="#">773282,4</a>	59 mins 39 secs	2021-02-19 02:07:24	SUCCESS
✔ openstack-tox-py38-arm64 (non-voting)	openstack/neutron	master	check-arm64	<a href="#">776567,1</a>	53 mins 53 secs	2021-02-19 02:09:37	SUCCESS
✘ openstack-tox-py38-arm64 (non-voting)	openstack/neutron	master	check-arm64	<a href="#">776568,1</a>	53 mins 39 secs	2021-02-19 02:09:38	FAILURE
✔ openstack-tox-py38-arm64 (non-voting)	openstack/neutron	master	check-arm64	<a href="#">772255,10</a>	53 mins 13 secs	2021-02-19 01:56:10	SUCCESS
✔ openstack-tox-py38-arm64 (non-voting)	openstack/ironic	master	check-arm64	<a href="#">776540,4</a>	10 mins 59 secs	2021-02-19 01:05:15	SUCCESS
✔ openstack-tox-py38-arm64 (non-voting)	openstack/nova	master	check-arm64	<a href="#">758918,8</a>	19 mins 56 secs	2021-02-19 00:51:48	SUCCESS
✔ openstack-tox-py38-arm64 (non-voting)	openstack/neutron	master	check-arm64	<a href="#">765840,7</a>	37 mins 55 secs	2021-02-18 22:41:48	SUCCESS
✔ openstack-tox-py38-arm64 (non-voting)	openstack/heat	master	check-arm64	<a href="#">767161,21</a>	13 mins 29 secs	2021-02-18 22:49:29	SUCCESS
✔ openstack-tox-py38-arm64 (non-voting)	openstack/ironic	master	check-arm64	<a href="#">776540,3</a>	11 mins 11 secs	2021-02-18 22:04:47	SUCCESS
✔ openstack-tox-py38-arm64 (non-voting)	openstack/keystone	master	check-arm64	<a href="#">759940,5</a>	38 mins 24 secs	2021-02-18 19:57:52	SUCCESS

Consistently checking job status is very essential as we hope to provide stable services across different environments. We didn't set any schedule to turn the job to voting until the job is stable and passed for most projects. So for now, we set two follow-up tasks: check and maintain this job and try to have more projects adopt this job while making sure the job stays stable and healthy.

### Python wheels support for ARM64

OpenStack has started support for building and publishing python wheels for non-x86 CPU architectures. [Zuul jobs](#) that



support building and publishing python wheels for the ARM64 architecture have been added (by Marcin Juszkiewicz and Ian Wienand) to OpenStack since May 2020.

```
44 - publish-wheel-mirror-debian-buster
45 - publish-wheel-mirror-ubuntu-focal
46 - publish-wheel-mirror-ubuntu-bionic
47 - publish-wheel-mirror-ubuntu-xenial
48 - publish-wheel-mirror-centos-7-arm64
49 - publish-wheel-mirror-centos-8-arm64
50 - publish-wheel-mirror-debian-buster-arm64
51 - publish-wheel-mirror-ubuntu-focal-arm64
52 - publish-wheel-mirror-ubuntu-bionic-arm64
53 - publish-wheel-mirror-ubuntu-xenial-arm64
54 - release-wheel-mirror:
55     dependencies:
56         - publish-wheel-mirror-centos-7
57         - publish-wheel-mirror-centos-8
58         - publish-wheel-mirror-debian-buster
59         - publish-wheel-mirror-ubuntu-focal
60         - publish-wheel-mirror-ubuntu-bionic
61         - publish-wheel-mirror-ubuntu-xenial
62         - publish-wheel-mirror-centos-7-arm64
63         - publish-wheel-mirror-centos-8-arm64
64         - publish-wheel-mirror-debian-buster-arm64
65         - publish-wheel-mirror-ubuntu-focal-arm64
66         - publish-wheel-mirror-ubuntu-bionic-arm64
67         - publish-wheel-mirror-ubuntu-xenial-arm64
```

The jobs run as periodic tasks to build, publish, and release ARM64 wheels for OpenStack CI mirrors. From this step, we should be able to have ARM64 specific python wheels packages from the OpenStack community directly. And our unit test job as discussed above or any other ARM64 python jobs will actually run on an ARM64 environment with python packages that have been built for ARM64 specifically. Currently, ARM64 is the only non-x86 CPU architecture with pre-built python wheels as we still need someone who is willing to donate environments for specific architectures before people who might be interested can jump in and add support for a specific CPU architecture.

### Scenario tests for ARM64 (All tests passed!)

We are adding a new [CI job](#) (by Kevin Zhao, Rico Lin, and Ian Wienand) in the OpenStack community that installs Devstack with services and runs a complete set of related tempest jobs on the ARM64 environment. As everything is now successfully installed and all tempest tests passed, I think we can officially say that OpenStack can be successfully running on Arm64.

```

- job:
  name: devstack-platform-arm64
  parent: tempest-full-py3
  description: arm64 platform tests
  nodeset: devstack-single-node-arm64
  timeout: 10800
  vars:
    devstack_localrc:
      IMAGE_URLS: https://github.com/cirros-dev/cirros/releases/download/0.5.2/cirros-0.5.2-aarch64-disk.img
      DEFAULT_IMAGE_NAME: cirros-0.5.2-aarch64-disk
      DEFAULT_IMAGE_FILE_NAME: cirros-0.5.2-aarch64-disk.img
      # For cirros over 0.5.0, VM instances running CirrOS should have at least 128 MB of memory to be usable
      # m1.nano (exactly 128) is not enough. Therefore, use m1.micro (192 MB).
      DEFAULT_INSTANCE_TYPE: m1.micro
      # Observed backup volume finished surpass 300 secs, and resuore volume surpass 600 secs, hence use 900 secs
      BUILD_TIMEOUT: 900
      WORKER_TIMEOUT: 180
      RABBIT_HEARTBEAT_TIMEOUT_THRESHOLD: 240
      RABBIT_HEARTBEAT_RATE: 8
      KOMBU_MISSING_CONSUMER_RETRY_TIMEOUT: 120
      RPC_CONN_POOL_SIZE: 60
      ENABLE_VOLUME_MULTIATTACH: False
      NOVA_RPC_RESPONSE_TIMEOUT: 120
      NEUTRON_RPC_RESPONSE_TIMEOUT: 120
      PIP_EXTRA_INDEX_URL: "https://{{ zuul_site_mirror_fqdn }}/wheel/{{ ansible_distribution | lower }}-{{ ansible_distribution_version }}-{{ ansible_architecture | lower }}"
      tempest_concurrency: 4
      devstack_local_conf:
        post-config:
          $NOVA_CONF:
            libvirt:
              cpu_mode: custom
              cpu_model: cortex-a57
              num_pcie_ports: 15

```

On the way to debug this job. We have fixed bugs and changed multiple configs.

#### Libvirt config

One detail you can find here is that we specify cpu model in Nova libvirt config because aarch64 does not have a default cpu model.

#### Kernel panic

We encountered a [kernel panic issue](#) when we used Cirros 0.5.0 as the image type for test instances, but the issue no longer exists once we move it to 0.5.2

#### reaches Timeout

We have to increase several timeout time config because those timeout time configs were designed for X86 at first place. In our observation, RPC and HTTP requests might reach the old timeout time.

#### Fix bugs

One mission for this test job is to make sure we continue to deliver usable services. During debugging the devstack job, We found and fixed bugs (like this [Nova Libvirt bug](#)). This means as long as we have this kind of job in OpenStack community, we can reach a better promise for OpenStack on Arm64.

#### Performance issue

It's usually common to allocate three to four times more CPU. Therefore we do use a flavor with larger CPU counts than the same tempest test task on x86 and hope to have the same performance. As an observation, current performance is still lower than in an x86 CI environment. And the overall running time is two times slower than x86. We still work on tuning the performance (like using PIP\_EXTRA\_INDEX\_URL to use the local mirror). Currently the job runs about 2.4-3 hrs and might take longer than 3hrs and reaches the overall timeout limit. As a comparison, usually the running time on x86 is around 1.2-2hrs. More work on tuning performance is coming.

If you have any idea or suggestion regarding this job and how we should move forward, please reply on the [mailing list](#) or join [our PTG](#) on.

And again, we appreciate every single CI resource that has been donated to OIF, so thank you Linaro and OSUOSL!

## Build OpenStack on your single node ARM64 right away

You can install OpenStack on your ARM64 environment in multiple ways, and all will succeed if you configure it correctly (and assuming that your hardware supports the installation). Here we show you how to install through Devstack. To run devstack and install OpenStack on your ARM64 environment, just download [Devstack](#), write the `local.conf` config file in the devstack repository, and finally run stack.sh. For general config file guidelines, please check this [document](#).

```
[[local|localrc]]
# This area is the min. requirement for devstack config
ADMIN_PASSWORD=secret
DATABASE_PASSWORD=$ADMIN_PASSWORD
RABBIT_PASSWORD=$ADMIN_PASSWORD
SERVICE_PASSWORD=$ADMIN_PASSWORD
#IPV4_ADDRS_SAFE_TO_USE=172.31.1.0/24v #FLOATING_RANGE=192.168.20.0/25
#HOST_IP=10.3.4.5

IMAGE_URLS=https://github.com/cirros-dev/cirros/releases/download/0.5.1/cirros-0.5.1-aarch64-disk.img
DEFAULT_IMAGE_NAME=cirros-0.5.1-aarch64-disk
DEFAULT_IMAGE_FILE_NAME=cirros-0.5.1-aarch64-disk.img
DEFAULT_INSTANCE_TYPE=m1.tiny
BUILD_TIMEOUT=300
ENABLE_VOLUME_MULTIATTACH=False
GLANCE_USE_IMPORT_WORKFLOW=True
DISABLED_SERVICES+=,s-account,s-container,s-object,s-proxy,c-bak
[[post-config|$NOVA_CONF]]
[libvirt]
cpu_mode=custom
cpu_model={Please put your CPU model here}
num_pcie_ports=15
```

You can also use Kolla, Helm, and other tools as well.

## Various ppc64le efforts

While much community effort is around aarch64, there is also some community effort around ppc64le. Nova has had PowerKVM 3rd-party CI for a long time, but the efforts go far beyond this. In particular, a lot of effort is in TripleO and RDO. TripleO has good support for deploying on ppc64le, especially for deploying a cloud that is a mix of x86\_64 and ppc64le. For the community, deploys using TripleO will use content from RDO, and there has been a lot of effort to get all RDO content working properly for ppc64le. It is still an ongoing challenge to publish RDO images (container images and disk images) for ppc64le, but this should be in place for the community very soon. Please reach out if you would like to help with getting things published, or would like help with building locally. And of course, there are packages in CentOS 8 for deploying an OpenStack cloud (TripleO-based or otherwise) on ppc64le (and aarch64 too!).

## Cross community efforts

### OpenStack and Kubernetes

We also promoted the multi-arch effort for [kubernetes/cloud-provider-openstack](#) (in the Kubernetes community) by adding [scripts](#) (by Rico Lin) to build and publish container images with multiple architectures.

We try to make sure CPU architecture support is really applied to users and their use cases. As Kubernetes is one of the most popular use cases on top of OpenStack, we took a step to join into the Kubernetes community too and try to make sure the multi-arch scenario works for Kubernetes on OpenStack. The first thing we took on is adding multi-arch support in the OpenStack cloud provider.

For those who don't know what a cloud provider is for Kubernetes is: it's a plugin mechanism that allows different cloud providers (OpenStack, AWS, GCE, etc) to integrate their platforms with Kubernetes, so an admin can manage those resources through cloud-controller-manager component to fulfill cloud-specific control logic.

This mechanism allows OpenStack to be more tightly integrated into Kubernetes projects. And to introduce multi-arch support for OpenStack cloud providers allows all community members to have certified images from community directly. It's not automatically built and published by the community yet, but we plan to have those multi-arch container images be automatically built and published by CI jobs in the future.

There's also a developer document to show you how to build OpenStack cloud provider container images for multiple

architectures, so even before the automation work in the community is done, you can follow the document and use the Makefile script to build and publish your own. Simply run:

```
export ARCHS='arm64'
make upload-images
```

This will help you build and publish all OpenStack cloud provider images for arm64 architecture to your container image repository. The command will build Go packages and add it with some required dependencies on top of basic images (most uses alpine:3.11, only cinder-csi-plugin uses debian-base:1.0.0). And it will build all the images you need for all the architectures you ask for. Finally you can push images through `docker push` (of course you can override docker to any other container engine you desire).

### OpenStack and OpenEuler

[OpenEuler](#) is the newest open source operating system, which is quite famous and widely used in China. It was born from Huawei and then quickly attracted quite a lot of companies and contributors, which made it a first-class-citizen in the open source OS community. It is rpm based and has support for x86 and aarch64 architectures. The integration between OpenStack and OpenEuler has mainly 2 parts:

- At the upstream level, there should be support for openEuler in OpenDev CI. This part means to support openEuler in devstack and pass tempest tests. Also, diskimage-builder should also support openEuler. Linaro has contributed some effort on this, and we've finished the POC and [submitted the patch](#) to devstack to support this function. The diskimage-builder support for building OpenEuler is a work in progress.
- At the release level, OpenEuler has set up an OpenStack SIG, which is in charge of releasing OpenStack packages. Quite a lot of dependencies have been fixed for OpenStack packages, and the SIG is working towards supporting OpenStack V version for OpenEuler 21.03.

### Efforts across projects

There already have been efforts for multiple architecture support across the OpenStack community.

- Kolla supports [jobs](#) (thanks to Marcin Juszkiewicz) for building and publishing container images for aarch64.
- Multiple patches under Nova, Neutron, etc, to support different CPU architectures. Like Nova libvirt driver have some fixes for non-x86 architecture because the driver itself is fully tested on x86 in community. So issues like `Fix check amd\_sev\_support on AArch64 [fixed]` happens. [Here](#) also mention a few previous fixes.
- [Magnum support Arm64](#) for the Fedora Atomic driver
- [Arm64 function](#) test for diskimage-builder
- A new pipeline "[check-arm64](#)" (thanks to Ian Wienand) in the community infrastructure to separate ARM64 CI workflow from x86.
- If you explore around the entire community (like if you search for "[aarch64](#)" or "[ppc64le](#)"), you will find more OpenStackers who fight for multiple architectures.

### Documentation

We have some [basic aspects](#) documented (Thanks to Jeremy Freudberg) to make some basic clear information- which will need to be improved once we have more progress or have people willing to update it. There are also some other documents like the [support matrix for Nova](#) to explain the functionality under different architectures. This matrix might not be the most up-to-date all the time (as missing features for some CPU architectures might get implemented before we update this matrix) but it still has some reference value.

### What's missing from OpenStack

Here we collect some missing resources/features and some other things to be aware of for non-x86 architectures

#### CI resources

The very first thing for community to support specific CPU architecture and be capable of building, testing, and publishing service is to have CI resources for it. Currently We only have two clusters from Linaro and OSUOSL for ARM64 environment. It's a good start for ARM64, but also means the rest of the non-x86 architectures will not get much guarantee from community (It does not mean it's less valuable or unusable, just not much community can do to have better support). Third-party CI is okay, but full community infrastructure is better.

#### Deployment tools supporting heterogeneous deploys

Many deployment OpenStack tools can deploy non-x86 just by pointing at different content, but very few (for sure, TripleO, as discussed above) can deploy for a mix of architectures. Due to cost or different interests it is useful to have a single cloud with more than one architecture.

#### Fully tested for multi-arch

ARM64 is the most tested non-x86 architecture we can get, but still missing functional test jobs (patch proposed but not landed yet) for running tempest tests on and still need to turn unit test jobs to voting.

## Summit Sessions

There were sessions for multi-arch in past summits. And the Multi-Arch SIG will try to keep providing presentations for multi-arch support status updates going forward. You can find most of the presentation videos [online](#) (like [Arm64 related videos](#))

Some recommended sessions for beginners:

- 2020 report from Multi-Arch SIG: [Multi-Arch Support in OpenStack](#)
- [The Progress for Cloud Computing on Arm Architecture](#) which includes analysis of current cloud with ARM architecture, some work share from Linaro, and business view about ARM64 support from EasyStack and Huawei. There's a [great article from Superuser](#) for this session by Kevin Zhao, Zhenyu Zheng, and ChangBo Guo. So don't miss that one too.
- [Three ARMed OpenStack - Building and Performance Testing OpenStack on ARM](#) provides some experiences around build and testing
- [Kata Containers on Arm Lets talk about our progress!](#) This session provides updates for Kata Containers on ARM

There are other great sessions that we didn't list here, like demo or marketing analysis sessions, so go ahead and find what suits you.

## More Community Efforts

There are some community efforts that we know from OpenStack User Groups around the world.

For example, the China User Group hosted some online meetups for technical/community exchange about Open Infrastructure with ARM64. They also have a Wechat group to share ARM-related information (Let us know if you wish to join this WeChat group: Rico Lin [ricolin@ricolky.com](mailto:ricolin@ricolky.com), or Horace Li [horace@openstack.org](mailto:horace@openstack.org)). There are also efforts about OpenEuler with OpenStack (as mentioned above).

Korea User Group made a study group for `Open Infrastructure with ARM` to share and study from each other's experiences or community documentation. They study by searching and brainstorming. And they received a sponsorship for ARM servers from an ARM server manufacturer (XSLAB Inc.)

We believe there are more user groups that have efforts in multi-arch support for infrastructure. So please share your story with us!

## Authors

- Rico Lin, EasyStack
- Jeremy Freudberg, Red Hat
- Kevin Zhao, Linaro
- Horace Li, Open Infrastructure Foundation for Translation