

# Bootstrap (Getting Started with ONAP Development)

That is a good question...

It will depend on what you want to do in ONAP.

The mission of the integration projects are very different. We develop test, deploy labs, setup CI/CD chains...so we need to get skills from different fields (dev/System/automation/...).

The main Integration page is a good first step, it references the documentation, repository, Q&A: [Integration Project](#)

More generally the wiki is full of lots of resources: [Developing ONAP](#) ...but it can be a bit misleading as everything is not maintained (Developing ONAP not updated since 2018)

The official documentation is also a good source: <https://docs.onap.org/>

All the bugs/tasks/Epics are tracked through a JIRA system. Code contributions are provided through a git/gerrit system. A basic knowledge of these tools is a good prerequisites.

If you want to be involved, it is recommended to join the weekly meeting (Wednesday 1PM UTC see [Integration Project](#). The agenda is open, you may add a topic, do not be shy to ask for a quick starting tour and explain what you would like to do.

## Developers

If you want to create new tests, you can use any framework you want. For the moment the 3 main frameworks used are:

- robotframework
- python\_onapsdk
- bash scripts

We may also mention some tests in GO..

All these tests are hosted in different places as some have been developed upstream, some by security team, some by integration team.

Robot tests are located in the repo: <https://gerrit.onap.org/r/admin/repos/testsuite>

Python tests leverage the [python\\_onapsdk](#) are located in the repo: <https://gerrit.onap.org/r/admin/repos/testsuite/pythonsdk-tests>

Other tests can be found in <https://gerrit.onap.org/r/admin/repos/integration> and other repositories

since Frankfurt we try to manage small repositories..so if you want to bring new tests, it has to be discussed with integration, but we may create a new repo if needed

## Integrators

### Packaging

We shall maintain the java dependencies through the oparent project: <https://gerrit.onap.org/r/admin/repos/oparent>

We must also build baseline images for java (<https://gerrit.onap.org/r/admin/repos/integration/docker/onap-java11>) and python (<https://gerrit.onap.org/r/admin/repos/integration/docker/onap-python>).

Projects can then integrate these images respecting seccom requirements and hosted in the nexus.

Integration is responsible to maintain them, fix security vulnerabilities (once for all the project using these images)

### CI/CD

We need to create and maintain CI/CD chains. See [Where can I see CI/CD ONAP status?](#) for details.

For this kind of activities, skills in jenkins, gitlab-ci, bash, system are welcome.

We shall provide an integration view to the Release management and verify that the dockers are delivered as expected in OOM for a given release.

Our CI/CD chains must also help release management to defined when the release is ready..or not.

The tests are embedded in xtesting dockers that are built through the LF system (<https://gerrit.onap.org/r/admin/repos/integration/xtesting>).

Integration shall maintain these dockers. So docker knowledge is not bad in this area as well...

### Integration pod

a testsuite (robot) pod is provided aside of the release to help launching test on a deployed ONAP.

Even if this pod is not used for the daily/weekly/gating, this pod is know widely used by the community to launch the healthcheck or more complex tests on a deployed ONAP.

Integration shall guarantee the validity of this pod.

## Lab administrators

We are providing lab resources for the projects [How can I get access to one of the ONAP labs?](#).

The main asset is the windriver/Intel lab.

We must clean, audit the performance of the DC and help the WIndriver/Intel team in case of upgrade/migration.

Recently we added a staging lab on Azure based on the same CICD chains than the daily/weekly/gating chains.

Skills in System/Openstack/Azure/lab admin are needed.

## Old references

- [Building & Testing an ONAP Component Locally](#)
- [Developing ONAP](#)
- [ONAP on Vagrant](#) - a way to deploy ONAP into a development environment
- [Vagrant Jenkins](#) - automatically sets up a Jenkins VM with pre-defined jobs to build all the ONAP Java code and Docker images