
developer.skatelescope.org

Documentation

Release 0.1.0-beta

Marco Bartolini

Feb 19, 2020

1	SKA Ansible Playbooks	3
2	Quick Start	5
2.1	Playbooks	5
2.2	Development Environment TESTED OS (using a box requires at least 4GB RAM):	6
3	Kubernetes and the MVP	7
3.1	Skampi	7
3.2	Ansible Playbook for local Kubernetes	7
4	Makefile	9
4.1	With Vagrant	10
4.2	Minikube with Vagrant	10
4.3	Clean up	10
4.4	Reset Minikube cluster	10
4.5	Minikube Direct	11
4.6	Minikube on kvm2	11
4.7	Clean up	11
4.8	Minikube direct install (for the brave)	11
5	Deploy Kubernetes Cluster	13
5.1	Set up N-node cluster	13
5.2	Join nodes to Kubernetes Cluster	13
6	GitLab Runner Environment	15
7	List of playbooks	17
7.1	Tango Development Environment	17
7.2	Minikube	17
7.3	Runners	17
7.4	Kubernetes clusters	17
7.5	Helm Linting	18
7.6	Openstack provisioners	18

This is the README that you can also find on the [Gitlab repository](#).

CHAPTER 1

SKA Ansible Playbooks

This repository consists of a Makefile and a set of playbooks that allow a developer to easily prepare working environments, be it bare metal or virtual machines, for

- working on TANGO-controls;
- deploying, testing and running the SKA MVP Integration project (SKA MPI)
- setting up Gitlab Runners
- creating and deploying Kubernetes clusters on Openstack
- Deploying a Prometheus service for monitoring resources
- Helm linting

... and more. Please visit our [Documentation](#) for details on all the playbooks.

CHAPTER 2

Quick Start

- Clone the repo:

```
git clone https://gitlab.com/ska-telescope/ansible-playbooks.git && cd ansible-  
↳playbooks
```

For working with Ansible, you need to install it and set up a connection (ssh if it's not localhost) to the machine(s) where the playbooks will run. Most of the `make` targets merely call one or more Ansible playbooks, so you need to install Ansible to use it too.

- Install Ansible:

```
apt-add-repository --yes --update ppa:ansible/ansible && apt-get install ansible -  
↳y
```

or with `sudo`:

```
sudo apt-add-repository --yes --update ppa:ansible/ansible && sudo apt-get -  
↳install ansible -y
```

- Edit the file “hosts” with the address(es) to want to manage (if not localhost).
- Add the ssh key to the managed hosts (if not localhost).

2.1 Playbooks

The `deploy_tangoenv.yml` playbook can be used for deploying a local Tango Controls development environment:

2.1.1 Deploy Tango Environment

To deploy a Tango development environment, you simply now need to launch the `ansible-playbook deploy_tangoenv.yml`:

```
ansible-playbook -i hosts deploy_tangoenv.yml
```

or with a password (replace \$PASSWORD with an actual password):

```
export PASSWORD=my-sudo-user-password
ansible-playbook -i hosts deploy_tangoenv.yml --extra-vars "ansible_become_pass=
↪$PASSWORD"
```

To work with pytango, activate the virtualenv:

```
source /venv/bin/activate
```

The following variables can be set:

```
build_tango: default('yes')
install_pytango: default('yes')
install_ide: default('yes')
install_ska_docker: default('yes')
start_tango: default('yes')
update_hosts: default('yes')
```

For example:

```
ansible-playbook -i hosts deploy_tangoenv.yml --extra-vars "build_tango='no' ↪
↪install_mysql='no' install_ide='no'"
```

For more information on the other playbooks available in this repo, visit the [Documentation](#).

2.2 Development Environment TESTED OS (using a box requires at least 4GB RAM):

- ubuntu:18.04
- ubuntu:16.04
- debian:stretch-slim

CHAPTER 3

Kubernetes and the MVP

The simplest way of deploying the MVP is by running it on a Minikube system. Minikube is a single-node cluster provisioned with all the basics that give a developer the opportunity to try out kubernetes.

3.1 Skampi

It is possible to install the integration environment locally with Minikube, as well as deploy the SKA MPI project (the MVP), by running the Ansible playbook or calling the make command:

```
make skampi
```

The make target above calls the playbook for setting up the skampi repository, with additional parameters already passed from the rules (or `PrivateRules.mak`, if you've set it up). It assumes password-less sudo rights.

Using an Ansible playbook, you may need to set a password, as in the example below; if you have password-less sudo access on the host you don't need that.

```
ansible-playbook -i hosts deploy_skampi.yml --extra-vars "ansible_become_pass=  
↪$PASSWORD"
```

At the following link will be setup the webjive webapplication: <http://integration.engageska-portugal.pt/testdb>

3.2 Ansible Playbook for local Kubernetes

The following are a set of instructions for deploying Kubernetes either directly locally or on a Vagrant VirtualBox. It has been tested on minikube v1.1.1 with Kubernetes v1.14.3 on Ubuntu 18.04, using Vagrant 2.2.4.

The aim of these instructions, scripts, and playbook+roles is to provide a canned locally available Kubernetes development environment. This environment will contain:

- Kubernetes at 1.14+ on Docker

- Tools: `kubectl` and `helm` configured with a local [Tiller-less Helm](#)
- A running Ingress Controller
- [Network Policies](#) implemented with Calico as the Pod network
- This project mounted into the Guest OS at `/vagrant`

CHAPTER 4

Makefile

All actions are available as targets in the Makefile - type `make help` to get the list of available targets, and variables that can be supplied at the command line. To make your own variables permanent, place them in a `PrivateRules.mak` file in the root of this project.

```
$ make
Makefile:help                show this help.
Makefile:k8s                 Which kubernetes are we connected to
Makefile:localip             set local Minikube IP in /etc/hosts file for_
↪Ingress $(INGRESS_HOST)
Makefile:minikube             Ansible playbook for install and launching Minikube
Makefile:skampi              Ansible playbook for install and launching_
↪Minikube and the Skampi project
Makefile:vagrant_down         destroy vagrant instance
Makefile:vagrant_install      install vagrant and vagrant-disksize on Ubuntu
Makefile:vagrantip           set Vagrant Minikube IP in /etc/hosts file for_
↪Ingress $(INGRESS_HOST)
Makefile:vagrant_up           startup minikube in vagrant
Makefile:vars                 Vagrant and DISPLAY variables

make vars (+defaults):
Makefile:DRIVER               true  ## Run Minikube via 'kvm2' driver (true) or
↪'none' (false)
Makefile:INGRESS_HOST         integration.engageska-portugal.pt ## Ingress HTTP_
↪hostname
Makefile:USE_NGINX            false ## Use NGINX as the Ingress Controller
Makefile:V_BOX                ubuntu/bionic64 ## Vagrant Box
Makefile:V_CPUS               2    ## Vagrant/Minikube no. CPU allocation
Makefile:V_DISK_SIZE          32GB  ## Vagrant/Minikube disk size allocaiton in_
↪GB
Makefile:V_GUI                false ## Vagrant enable GUI
Makefile:V_IP                 172.16.0.92 ## Vagrant private network IP
Makefile:V_MEMORY             4096  ## Vagrant/Minikube memory allocation in MB
Makefile:V_PLAYBOOK           deploy_tangoenv.yml ## Ansible playbook run in_
↪Vagrant
```

(continues on next page)

(continued from previous page)

```
Makefile:XAUTHORITYx      ${XAUTHORITY}
```

4.1 With Vagrant

Install Vagrant and VirtualBox - on Ubuntu 18.04+ use:

```
apt install virtualbox vagrant
```

and then:

```
make vagrant_install
```

This will ensure that Vagrant is at minimally working level, and that the Vagrant plugin `vagrant-disksize` is installed which is required for the guest base box disk resizing.

There are two tested options for Vagrant base boxes - `ubuntu/bionic64`, and `fedora/29-cloud-base`. These can be supplied by var `V_BOX`.

Adjust the `vcpus`, `memory`, and initial disk size with: `V_CPUS`, `V_MEMORY`, and `V_DISK_SIZE` as above.

4.2 Minikube with Vagrant

Once Vagrant and VirtualBox are installed, launching a guest OS and installing Minikube with Ansible is carried out with the following:

```
make vagrant_up
```

The default action is to run the `deploy_tangoenv.yml` playbook creating a virtual machine named `tango-dev`. To create a vanilla Minikube+Vagrant system, add vars `V_NAME=minikube-vm` `V_PLAYBOOK=deploy_minikube.yml`.

Once successfully completed, inspect Minikube by `ssh`'ing onto the box with `vagrant ssh`, where all the usual `kubectl` capabilities are available:

```
kubectl get all --all-namespaces
```

The repository has been shared into the guest OS in `/vagrant` where the associated Helm Charts and `make` commands are available.

4.3 Clean up

Clean up with:

```
$ make vagrant_down
```

4.4 Reset Minikube cluster

Kubernetes environment can be reset using the following command:

```
ansible-playbook reset_k8s.yml
```

4.5 Minikube Direct

4.6 Minikube on kvm2

Minikube can be installed onto your Debian or RedHat based machine within a kvm2 virtual machine instance with:

```
make minikube DRIVER=kvm2
```

4.7 Clean up

Clean up with:

```
$ minikube delete
```

4.8 Minikube direct install (for the brave)

Minikube can also be installed directly onto your Debian or RedHat based machine with:

```
make minikube DRIVER=none
```

WARNING This will overwrite anything that you have locally installed for `docker`, `helm`, `kubectl`, and `minikube` which could be disastrous if you have an existing and customised configuration.

Deploy Kubernetes Cluster

A fully-fledged Kubernetes cluster can be deployed using the following command, BUT make sure `hosts` file is modified according to your needs:

5.1 Set up N-node cluster

After setting up the `hosts`, you can run the following playbook to provision the nodes.

```
ansible-playbook -i hosts setup_cluster.yml
```

This playbook is intended for CentOS 7 operating system. If deployed within the EngageSKA cluster, please create all nodes with `int_net` network and add to each VM floating IP. After the installation, remove floating IP's only from the worked nodes.

5.2 Join nodes to Kubernetes Cluster

Kubernetes worker can be joined to the cluster using the following command, BUT make sure `hosts` file is modified according to your needs:

```
ansible-playbook -i hosts join_cluster.yml
```

GitLab Runner Environment

It is possible to install the gitlab runner environment locally. Make sure the docker environment is installed and it has at least 50GB disk space.

Call the playbook with the following command:

```
ansible-playbook -vvv deploy_runners.yaml \
  --extra-vars "token='<token from your gitlab repository>' \
                name='runnerXXX' \
                taglist='tag1,tag2,tag3'" \
  -i hosts
```

- Get your token when you register the runner - go to [Settings >> CI/CD](#). NOTE: you will only be able to do this if you are a Maintainer of the project!
- Tags are the way that the runner is linked to jobs that are specific for this type of runner, i.e. `shell`, `docker-executor`, etc.
- For more info, go to <https://docs.gitlab.com/runner/register/>.

To check the gitlab-runner is working, go into gitlab project page and check the CI/CD settings - a status icon will show up.

List of playbooks

This is just a short list of all the currently available playbooks. Not all of them are “production ready” yet so are to be used with caution, and are not well-documented.

7.1 Tango Development Environment

- `deploy_tangoenv.yml`

7.2 Minikube

- `deploy_minikube.yml`
- `reset_k8s.yml`
- `deploy_skampi.yml` - currently deploys skampi on a Minikube system only.

7.3 Runners

- `deploy_runners.yml`
- `deploy_openstack_runner.yml`

TODO: above two playbooks need to be merged.

- `clean_runners.yml`

7.4 Kubernetes clusters

- `setup_cluster.yml` - create a cluster as specified in the `hosts` file

- `join_cluster.yaml` - adding a node to the cluster

7.5 Helm Linting

- `helm-linting.yaml`

7.6 Openstack provisioners

- `openstack.yaml`