

Kubernetes at SKA

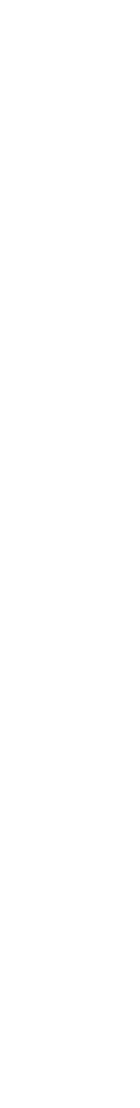
Matteo Di Carlo (INAF-OAAB) TANGO-Workshop @ICALEPCS 2021



About me

- Matteo Di Carlo (matteo.dicarlo@inaf.it)
- Working for INAF-OAAB since 2014
- Since 2015 in the SKA project
- Software engineer, in SKA part of the system team and coordinator of the cop-tango community
- https://orcid.org/0000-0002-3903-9637





ska-tango-examples repository

Demonstrates how to structure a project that pro simple Tango devices coded in PyTango

- Development and testing done
- Environments and test independent of host envire
- Many authors:
 - https://gitlab.com/ska-telescor
- The code shown in this presentation telescope/ska-tango-examples/-/tre









Requirements

- Install docker:
 - Follow the instructions available at ttps://docs.docker.com/get-docker/
- Install minikube:
 - <u>telescope/sdi/deploy-minikube/</u>
- Optionally, install host OS dependencies:
 - controls/cppTango/-/blob/main/INSTALL.md

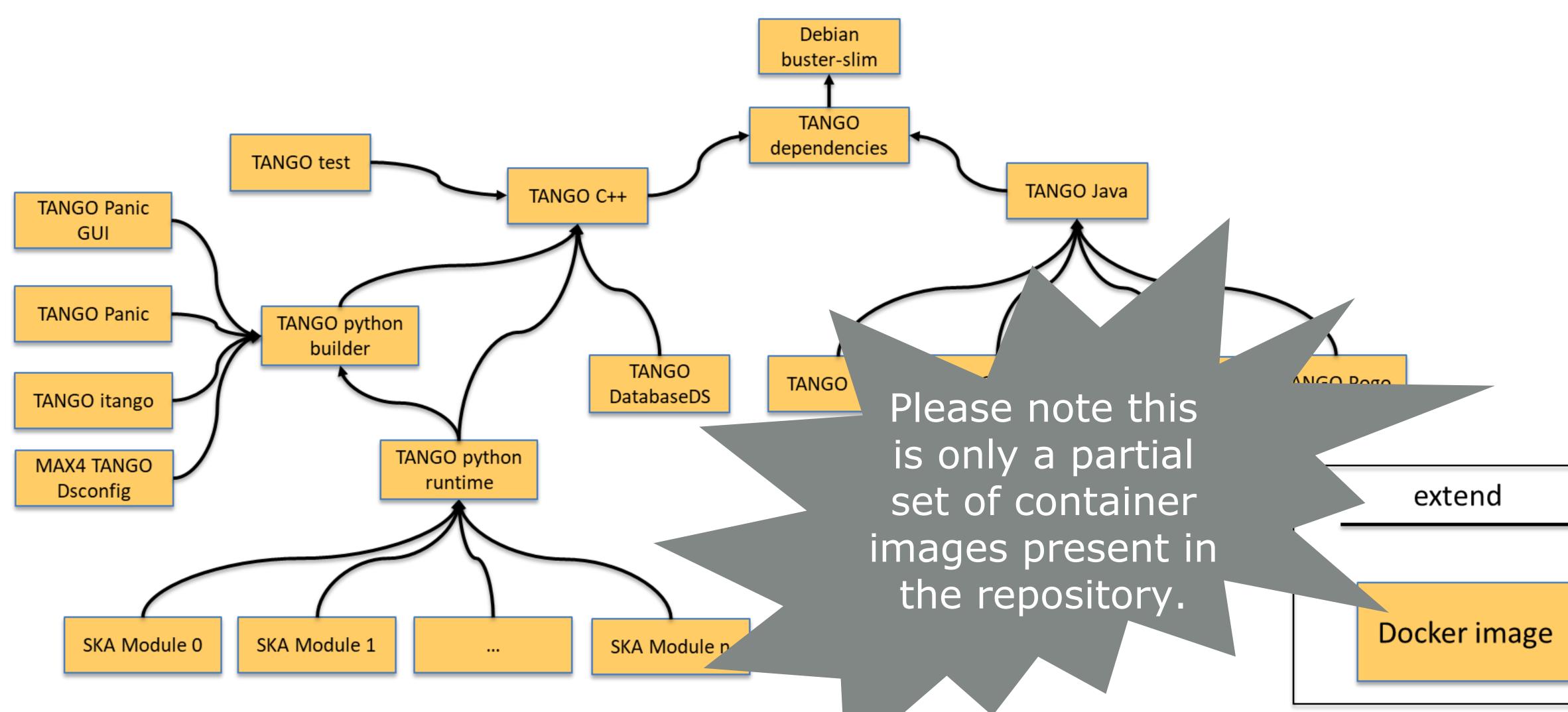


Follow the instructions available at https://gitlab.com/ska-

Compile the TANGO framework: https://gitlab.com/tango-



SKA-tango-images - Containerized environment for **TANGO-controls** application









Kubernetes and Helm

- Kubernetes (k8s) for container orchestration (kg)
 - Service == TANGO Device Server
- Helm for packaging SKA k8s applications
 - Tool for managing Kubernetes charts
 - Chart is a package of pre-configure information for running a Kube

For each SKA eler and there must be an helm chart for running it in k8s!

Use of Makefiles for lifecycle management (one command for build images, start application using helm, test application and clean)!



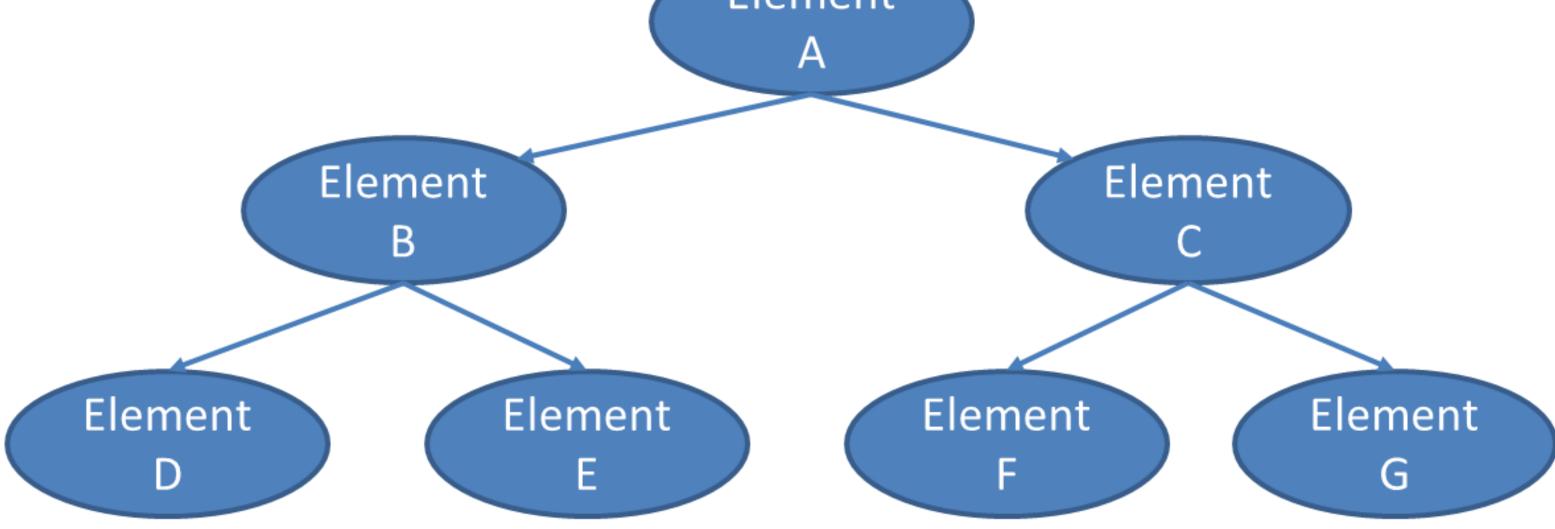
We are working in this area at the moment for standardising a common set of Makefile targets. This will bring every repository to have a standardized structure (set of folder, common files, etc.)



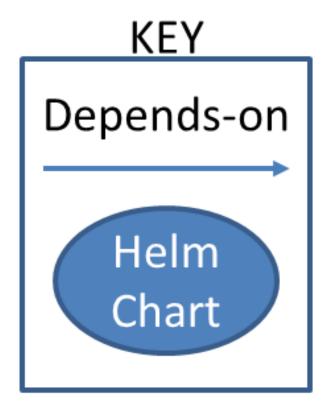
Architecture for integration (with Helm)

- Helm has the concept of dependency
 - An helm chart can have one or more sub-charts
- The integration of SKA elements can be done with this concept

Element









Umbrella charts

 Operational as charts For every SKA element, there is at least an • SO umbrella chart for integration testing

using dependencies: the subunen ame; a G ner ¹er. Α UMBRELLA R D











Development of a device step 1: POGO

- make start_pogo
 - Remember to export the DISPLAY and Xauthority environment variables
- Generate the device

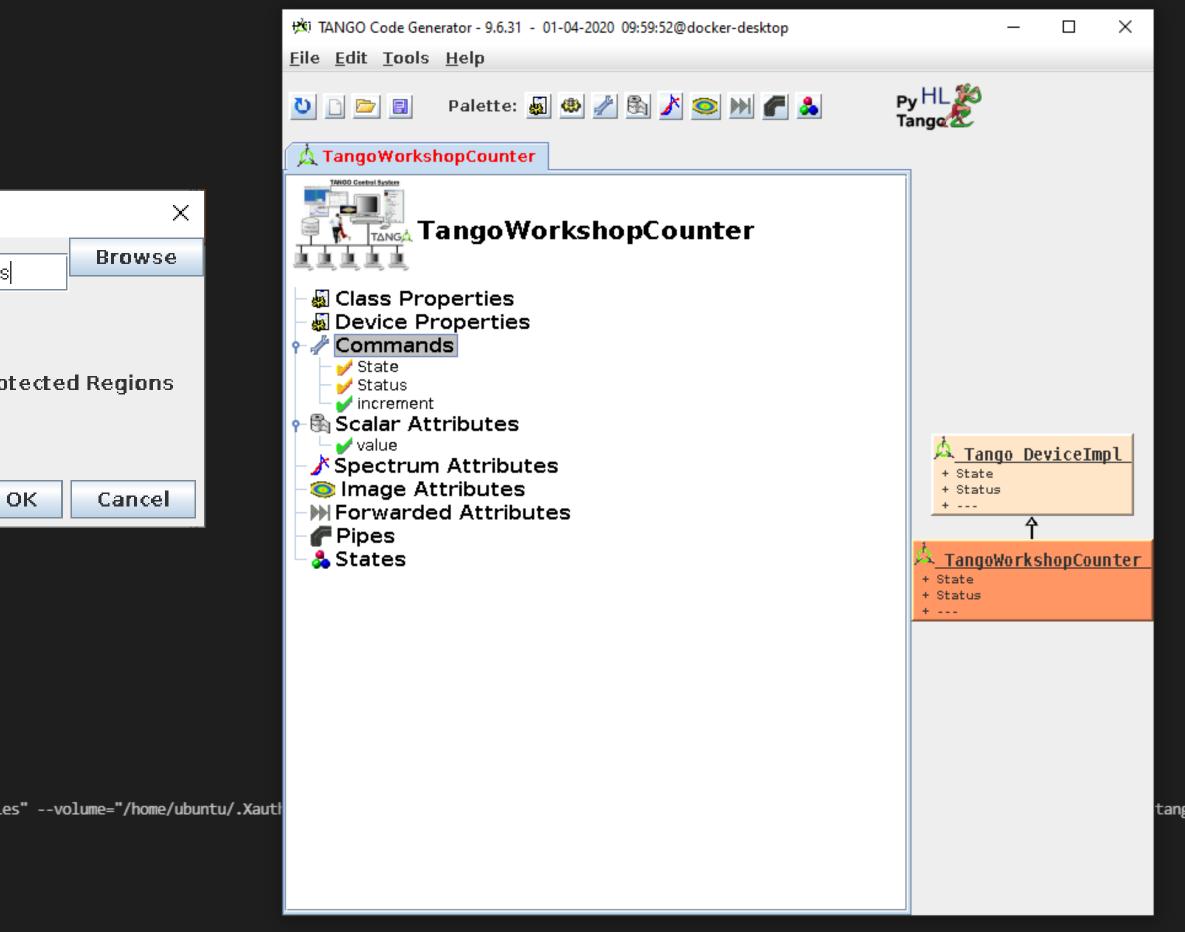


In the home folder there folder where you started Pogo





×1 F	ile Edit Selection View Go Ru	n Terminal Help		Makefile	e - CENTRALNO
Ð	PROBLEMS OUTPUT DEBUG CONS	OLE TERMINAL			
	<pre>.make/k8s.mk:TANGO_HOST .make/oci.mk:CAR_OCI_REGISTRY_ .make/oci.mk:OCI_BUILDER .make/oci.mk:OCI_BUILD_ADDITIO .make/oci.mk:OCI_IMAGE .make/oci.mk:OCI_IMAGES_TO_PUB .make/oci.mk:OCI_IMAGE_BUILD_C .make/oci.mk:OCI_IMAGE_FILE_PA</pre>	<pre>docker ## Image builder e NAL_ARGS ## Additional bui \$(PROJECT_NAME) ## Defaul \$(OCI_IMAGE_DIRS) ## Imag LISH \$(OCI_IMAGES) ## Imag ONTEXT . ## Image build co TH Dockerfile ## Image rec</pre>	g: docker, or podman ld argument string t Image (from /images/ es to lint and build es to publish ontext directory, rela ipe file	<pre>/<oci_image dir="">) ative to /images/<image di<="" pre=""/></oci_image></pre>	
Γø	.make/oci.mk:OCI_SK	registry.gitlab.com/pipel ration Preference Window@		11:0.10.0	
₽ ₽	.make/oci.mk:OCI_TO .make/python.mk:PYTI .make/python.mk:PYTI .make/python.mk:PYTI .make/python.mk:PYTI	Path : /home/tango/s	ka-tango-examples	/src/ska_tango_examp	les/teams
	.make/python.mk:PYT	to be generated i			
.	.make/python.mk:PYTI .make/python.mk:PYTI .make/python.mk:PYTI	to be generated : Device Class:	XMI file	Code files	Pro
))	<pre>.make/python.mk:PYTI .make/python.mk:PYTI</pre>		0.1.1.5	O Python Package	
*	<pre>.make/python.mk:PYT .make/release.mk:RE Makefile:CI_ENVIRON Makefile:CI_PROJECT</pre>	Documentation:	🔾 html Pages	Sphinx	
	<pre>Makefile:KUBE_NAMESPACE Makefile:RELEASE_NAME Makefile:UMBRELLA_CHART_PATH Makefile:WEBJIVE Makefile:XAUTHORITY (venv) ubuntu@LAPTOP-5LBGJH83: docker runnetwork hostus .32 Display is 172.25.16.1:0.0 Starting Pogo Appli under Linu Reading properties from file:/ Reading site properties from / /home/tango/.pogorc doc_home: ./doc_html makefile_home: /usr/local/sha install_home: \$(TANGO_HOME)/ site.name: null</pre>	<pre>\$(THIS_HOST):0 test-parent H event-generator ska-tang false# Enable jive /etc/deploy/config ## KUB ## base64 encoded kubect ska-tango-examples test charts/\$(HELM_CHART)/ false# Enable Webjive \$(HOME)/.Xauthority ~/ska-tango-examples\$ make er 1000:1000volvme="/ho x. usr/local/share/java/Pogo- usr/local/share/pogo/preferences</pre>	ECONFIG location l credentials for KUBE start_pogo me/ubuntu/ska-tango-ex 9.6.31.jar!/Pogo.defau	amples:/home/tango/ska-ta ult_properties	ngo-example
8	copyright:				
ŝ	******** Pogo GUI Release : 9			gorc (No such file or dire	ctory)
	: Ubuntu-20.04 🐉 a79b3b38 🗘 🛆				
	·片 🗳 S 🕀 🗙	📮 🕗 🤮 🦌	de la companya de la c		





Development of a device step 2: coding & testing

- Create a virtualenv
- Write your device and create a test (test-drive approach)
- Remember to lint:
 - make python-format && make python-lint
- Use DeviceTestContext or MultiDeviceTestContext
 - make python-test





∢	File Edit	Selection	View	Go Run	Terminal	Help				test_workshop.py	- CENTR	ALNODE
ۍ	EXPLO	RER				PROBLEMS 5 C	UTPUT	DEBUG CONSOLE	TERMINAL			
	✓ OPEN	EDITORS										
0	G	README.m	nd ska-tar	ngo-example	s							
\sim		1 Makefile sl			м	DEBUG tango:s PASSED	erver.py	/:1353 server]	loop exit			
0~		TangoWork				tests/unit/test_	workshor	o.pv::test incr	ement			
ိုင္ခ်ိ		1 Makefile sk										
		test_works				DEBUG tango:s	erver.py	/:1347 server]	loop started			
⇒a		RALNODE (WO										
		Timer.xmi	KKSPACE		110-20.04]	DEBUG tango:s	erver.pv	/:1353 server l	loop exit			
L <u>o</u>	4	WebjiveTest	Device.p	/		PASSED						
-0		WebjiveTest										
-0		_initpy										-
B		 DevFactory.py	v									
	√ te		y			report saved to:	build/r	reports/report.	json			
Ā									-			
		pycache				cove	rage: p]	latform linux,	python 3.7.			
		integration				Name					Stmts	Miss H
		unit				src/ska_tango_ex	amples/D	DevFactory.py			17	4
		pycache				src/ska_tango_ex					0	0
8		initpy				<pre>src/ska_tango_ex</pre>				r.py	48	24
~	4	test_calenda	ar_clock.p	y .		<pre>src/ska_tango_ex</pre>					47	7
8	4	test_counter	r.py			<pre>src/ska_tango_ex src/ska_tango_ex</pre>				ov	0 39	0 7
	4	test_event.p	у			src/ska_tango_ex				- 7	46	, 8
	4	test_long_ru	unning_de	evice_client.	ру	src/ska_tango_ex					0	0
	4	test_long_ru	unning_de	evice.py		src/ska_tango_ex					182	128
		test_motor.p	_			5, 371, 377-384,				446, 452	47	47
		test_power_		,		<pre>src/ska_tango_ex src/ska_tango_ex</pre>					17 5	17 0
		test_tabata.		, 		src/ska_tango_ex		-			174	69
		test_worksh			3, U	439-440, 445						
		_initpy				<pre>src/ska_tango_ex</pre>					0	0
		py conftest.py				src/ska_tango_ex 359, 364, 368	ampies/t	teams/CalendarC	лоск.ру		161	30
						src/ska_tango_ex	amples/t	teams/LogTestDo	wnstream.pv		9	9
		requirements	XL			<pre>src/ska_tango_ex</pre>					36	36
	> ve					<pre>src/ska_tango_ex</pre>					140	140
		ockerignore				src/ska_tango_ex					362	104
		ake8				485, 496-498, 5 src/ska_tango_ex					692-6 129	94, /12, 66
		itignore				, 383	ampres/ (compretor	Summingbev.	ccorrectly		00
	e. 🦊	itlab-ci.yml				src/ska_tango_ex		_		.ру	20	2
	e. 🧇	itmodules				<pre>src/ska_tango_ex</pre>			e.py		73	73
	.⊒	ylintrc				<pre>src/ska_tango_ex src/ska_tango_ex</pre>			stDevice py		97 128	97 128
	≣.re	elease				src/ska_tango_ex					128	128 0
	≡ AI	PACHE2-LICE	NSE			Bo_c/						
	≣ CI	HANGELOG.r	st			TOTAL					1734	949
8	≣ C0	OPYRIGHT				Coverage HTML wr				1		
		ockerfile				Coverage XML wri	cten to	Tile build/rep	orts/code-co	overage.xm1		
£63	> OUTLI											
202	> TIMEL	INE				(venv) ubuntu@LA	PTOP-5LE	BGJH83:~/ska-ta	ango-example	5\$		
→ w	SL: Ubuntu-	20.04 🦻 ST	T-982* ·	Operation Python	3.8.5 64-bit	'venv': venv) 🛛 🛞 0 🛆	5					
. =	H	-			-	1						
		<u> </u>	<u> </u>			-						

E (Workspace) [WSL: Ubuntu-20.04] - Visual Studio Code			—	۵
		2: bash (ska-tango-exar \smallsetminus	+~ 🗉]
live log teardown				
	[98%]			
live log setup				
live log teardown				
[[100%]			
ted json file: /home/tango/ska-tango-examples/build/reports/cucumber.json				

65% 39->42, 43-48 The command «make python-test» run on the host machine. Some tests can fail depending on the OS (i.e. windows). If it fails, the alternative is to run it on a container which won't fail. The command is «make pipeline_unit_test>>

43%

Branch BrPart Cover

68 passed, 3 deselected in 54.83s =

Ln 2, Col 1 Spaces: 4 UTF-8 LF Python 🔊 へ 幅 🥼 🕬 15:48



Development of a device step 3: deployment

- parent, used for testing.
- servers



 In order to install the examples, two charts have been created: one called ska-tango-examples which is the real application and the umbrella chart, called test-

 The ska-tango-examples uses the ska-tango-base chart for setting up the TANGO eco-system (mysql database and databaseds device) and the ska-tango-util library chart which helps in the definition of the TANGO device

More information on: https://gitlab.com/ska-telescope/ska-tango-images





How does it work?

- Define the device server and devices in a yaml file inside the folder data according to the documentation
- Reference that file in the main values file
- Use it in the template deviceservers.yaml
- It is possible to define <u>dependency</u> which will create one init container for each of them





Slide / 15

∢	<u>F</u> ile	<u>E</u> dit	Selection	<u>V</u> iew	<u>G</u> o	<u>R</u> un	<u>T</u> erminal	<u>H</u> elp			work	shopcoun	ters.yaml -	ska-tango
Ŋ	!	work	shopcounte	ers.yaml	u x	!	values.yam	М						
	c	charts > ska-tango-examples > data > ! workshopcounters.yaml												
0		1	_				s-{{.Rele							
\sim		2					amples-co							
~		3	domain:	ska-ta	ango-	exam	oles							
ငိုဝ ဝြ		4	instance	es: ["v	vorks	hopco	ounters"]							
		5	entrypo:	ints:										
æ		6	- name	e: "Tar	ngoWo	rksh	opCounter	.Tango	oWorkshopC	Counter"				
æ		7	patl	h: "/ap	op/sr	c/ska	a_tango_e	example	es/teams/T	angoWorks	hopCour	nter.py"	•	
_		8	server:											
Ľ		9	name:	"tabat	ta"									
		10	insta	nces:										
₿		11	– na	ame: "v	vorks	hopco	ounters"							
		12	c.	lasses:	:									
		13	- -	name:	"Cou	nter'							\triangleleft	<u>F</u> ile <u>E</u> dit <u>S</u> e
Ä		14		device	25:								<u>ل</u>	EXPLORER
		15		- name	e: "t	est/\	workshop_	counte	er/1"				\sim	
حيلته.		16		- name	e: "t	est/1	workshop_	counte	er/2"				Q	✓ SKA-TANG > build
•		17		prop	perti	es:							የօ	 ✓ ska-ta
		18		– na	ame:	"pol	led_attr'						9 o 0 o	> chart
		19		Va	alues	:							∠, œ	✓ data
		20		-	"pol	led_v	/alue"						22	! asy
Ś		21		-	"{{	.Valu	ues.devid	eServe	ers.worksh	nopcounter	s.polli	ing }}"		! cale {} cor
~		22	_depends_	_on:									_0	l eve
		23	- dev:	ice: sy	/s/da	taba	se/2						B	! for
		24	image:										π	! log
		25	regist	try: "{	[{ .V a	lues	.tango_e>	ample.	.image.reg	gistry}}"			囚	log mo
		26							age.image}	}"				! mu
		27					o_example							! pov
		28	pullPo	olicy:	"{{.	Value	es.tango_	_examp]	le.image.p	oullPolicy	}}"			! tab
													~	! the ! tim
														! wel
														! woi
														> temp ≣ .helm
														≡ .nein ≣ Char
														l Char
													8	! value
0													~~	> test-da > OUTLINE
Ø														> TIMELINE
2													× ws	SL: Ubuntu-20.04
క దికే														

Development of a device step 3: deployment

-examples [WSL: Ubuntu-20.04] - Visual Studio Code



đ



t Colorian Manu Co Dua Transiani			_
t <u>S</u> election <u>V</u> iew <u>G</u> o <u>R</u> un <u>T</u> erminal	<u>H</u> elp values.yaml - ska-tango-examples [WSL: Ubuntu-20.04] - Visual Studio Code	_	
PRER ····	! workshopcounters.yaml U ! values.yaml M ×		
EDITORS	charts > ska-tango-examples > ! values.yaml		
TANGO-EXAMPLES [WSL: U 🔓 🛱 ひ 🗗	89 instances: ["test"]		
uild	90 file: "data/logtestupstream.yaml"		A second se
ka-tango-examples	91		Non-
charts	92 logtestdownstream:		Non Maria and Noncorre Noncorre Noncorre
data •	93 instances: ["test"] 94 file: "data/logtestdownstream.yaml"		The Second secon
asynctabata.yaml	95		The Continue of the Continue o
calendarclock.yaml	96 multidevice:		Terrera Maria and Andreas
configuration.old.json	97 instances: ["test"]		
eventreceiver.yaml	98 file: "deta/multidevice.yeml"		
forattrtabata.yaml	99		
logtestdownstream.yaml	100 workshopcounters: 101 instances: ["workshopcounters"]		
logtestupstream.yaml	102 file: "data/workshopcounters.yaml"		
motor.yaml	103 polling: 1000		
multidevice.yaml	104		
powersupply.yaml	105 nodeSelector: {}		
tabata.yaml	106		
theexample.yaml	107 affinity: {}		
timer.yaml	108 109 tolerations: []		
webjivetestdevice.yaml	110		
workshopcounters.yaml U			
templates			
.helmignore			
Chart.lock			
Chart.yaml			
values.yaml M			
est-parent INE			
INE			
-20.04 🖇 ST-982* 😯 Python 3.8.5 64-bit	t ('venv': venv) 🛞 0 🔬 0 Ln 104, Col 1 Spaces: 2 UTF-	8 LF YA	AML



Development of a device step 4: install on minikube

«make oci-build» builds the container image (with docker)

«make install-chart» installs the deployment

Name:	ska-tango-examples
Labels:	<none></none>
Annotations:	<none></none>
Status:	Active

No resource quota.

No LimitRange resource.

install-chart: install charts/test-parent/ release: test in Namespace: ska-tango-examples with params: --set global.minikube=true --set global.tango_host=tango-host-databaseds-from-makefile-test:10000 --set ska-tango-base.display=172.25.29.163:0 --set ska-tango-base.xauthority=/home/ubuntu/.Xauthority --set ska-tango-base.jive.enabled=false --set webjive.enabled=false --set tango_example.tango le.image.tag=0.4.15-dirty --set event_generator.events_generator.image.tag=0.4.15-dirty --values gilab_values.yaml helm upgrade --install test \

--set global.minikube=true --set global.tango_host=tango-host-databaseds-from-makefile-test:10000 --set ska-tango-base.display=172.25.29.163:0 --set ska-tango-base.xauthority=/home/ubuntu/.Xauthority --set ska-tango-base.jive.enabled=false --set webjive.enabled=false --set tango_example.tango_example.image.tag=0.4.15-dirty --set event_generator.events_generator.image.tag=0.4.15-dirty --values gilab_ values.yaml \

charts/test-parent/ --namespace ska-tango-examples; \ rm gilab values.yaml

Release "test" does not exist. Installing it now. NAME: test

LAST DEPLOYED: Mon Oct 11 16:04:58 2021

NAMESPACE: ska-tango-examples

STATUS: deployed **REVISION: 1**

TEST SUITE: None

(venv) ubuntu@LAPTOP-5LBGJH83:~/ska-tango-examples\$











Development of a device step 4: wait and watch

«make wait» wait for all pods to be running «make watch» to see what's happening

×] <u>F</u> i	le <u>E</u> dit <u>S</u> election <u>V</u> iew <u>G</u> o <u>R</u> un <u>T</u> erminal <u>H</u> elp		workshopcounter	rs.yaml - ska-tan	go-examples [WSL: Ubuntu-20.04] - Visual Studio) Code	- @ ×				
Сŋ	PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL					× 1 <u>E</u>	ile <u>E</u> dit <u>S</u> election <u>V</u> iew <u>G</u> o <u>R</u> un <u>T</u> erminal <u>H</u> elp			workshope	counters.yaml - ska-ta
ρ	Every 2.0s: kubectl get all,pv,pvc,ingress -n s	ska-tango	-examples				PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL				
\sim	Warning: extensions/v1beta1 Ingress is deprecat	ted in v1	.14+. unavailable	in v1.22+:	use networking.k8s.io/v1 Ingress	\cap	<pre>device-server => artefact.skao.int/ska-tang</pre>	o-exampl	es:0.4.15-di	rty	
	NAME	READY	STATUS	RESTARTS		Q	Ded. workshopsourtons config ultur				
Ϋ́ο ο	pod/asynctabata-config-648kj	0/1	PodInitializing		18s		Pod: workshopcounters-config-wltkj Containers:				
	pod/asynctabata-tabata-0	0/1	Init:0/2	0	18s		dsconfig => artefact.skao.int/ska-tango-ima	Jec-tang	o-dsconfig.1	5 1	
	pod/calendarclock-config-fs2vj	0/1	PodInitializing	0	19s	06		ses cang	o uscomig.i		
a⊂ _	pod/calendarclock-test-0	0/1	Init:0/2	0	18s		Pod: workshopcounters-workshopcounters-0				
~	pod/event-generator-test-0	1/1	Running	0	19s		Containers:				
	pod/eventreceiver-01-0	0/1	Init:0/2	0	17s	~	<pre>device-server => artefact.skao.int/ska-tang</pre>	o-exampl	es:0.4.15-di	rty	
	pod/eventreceiver-config-z5b47	0/1	PodInitializing		18s						
	pod/fatabata-config-pmkpj	0/1	PodInitializing		19s	L _O	Mon Oct 11 16:06:05 CEST 2021				
ß	pod/fatabata-test-0	0/1	Init:0/7	0	17s		(venv) ubuntu@LAPTOP-5LBGJH83:~/ska-tango-e	kamples\$			
	pod/logtestdownstream-config-ljn8q	0/1	PodInitializing		19s	ß	NAME	READY	STATUS	RESTARTS	AGE
	pod/logtestdownstream-test-0	0/1	Init:0/2	0	18s	ш	asynctabata-config-648kj	0/1	Completed		2m22s
囚	pod/logtestupstream-config-4x9tg	0/1	PodInitializing		18s		asynctabata-tabata-0	1/1	Running	0	2m22s
	pod/logtestupstream-test-0	0/1	Init:0/2	0	17s	四日	calendarclock-config-fs2vj calendarclock-test-0	0/1		0	2m23s
	pod/multidevice-config-hxgqg	0/1	PodInitializing	0	19s		event-generator-test-0	1/1 1/1	Running Running	0 0	2m22s 2m23s
	pod/multidevice-test-0	0/1	Init:0/2	0	17s	-1	eventreceiver-01-0	1/1	Running	0	2m235 2m21s
	pod/ska-tango-base-itango-console	0/1	Init:0/1	0	19s		eventreceiver-config-z5b47	0/1	Completed		2m215 2m22s
	pod/ska-tango-base-tangodb-0	1/1	Running	0	19s		fatabata-config-pmkpj	0/1	Completed		2m23s
	pod/tabata-config-5d225	0/1	PodInitializing	â	195 19s		fatabata-test-0	1/1	Running	õ	2m21s
	pod/tabata-counters-0	0/1	Init:0/2	0	18s		logtestdownstream-config-ljn8q	0/1	Completed		2m23s
23	pod/tabata-tabata-0	0/1	Init:0/2	0	185 18s		logtestdownstream-test-0	1/1	Running	0	2m22s
	pod/tango-host-databaseds-from-makefile-test-0		Running	0	185 185		logtestupstream-config-4x9tg	0/1	Completed	0	2m22s
	pod/tangotest-config-bjbhm	0/1	PodInitializing		185 18s		logtestupstream-test-0	1/1	Running	0	2m21s
	pod/tangotest-test-0	0/1	Init:0/2	0	19s		multidevice-config-hxgqg	0/1	Completed		2m23s
	pod/theexample-config-2sdpt	0/1	PodInitializing		195 19s		multidevice-test-0	1/1	Running	0	2m21s
	pod/theexample-test-0	0/1	Init:0/2	0	18s		ska-tango-base-itango-console	1/1	Running	0	2m23s
		0/1	PodInitializing	9	18s		ska-tango-base-tangodb-0	1/1	Running	0 0	2m23s
	pod/timer-config-jnnjj pod/timer-counters-0	0/1	Init:0/2	0	17s		tabata-config-5d225 tabata-counters-0	0/1 1/1	Completed		2m23s 2m22s
	pod/timer-timer-0	0/1	Init:0/2	0	175 17s		tabata-tabata-0	1/1 1/1	Running Running	0 0	2m22s 2m22s
8	pod/webjivetestdevice-config-2m99t	0/1	Init:0/1	0	17S 18s		tango-host-databaseds-from-makefile-test-0	1/1	Running	0	2m22s
	pod/webjivetestdevice-test-0	0/1	Init:0/2	0	18s		tangotest-config-bjbhm	0/1	Completed		2m22s
572	pod/webjivetestdevice-test-0 pod/workshopcounters-config-wltkj	0/1	PodInitializing		18s		tangotest-test-0	1/1	Running	0	<u>2m23s</u>
- 5 <u>6</u> <u>7</u>	pod/workshopcounters-workshopcounters-0	0/1	Init:0/2		19s		theexample-config-2sdpt	0/1	Completed	0	2m23s
			-	0	195		theexample-test-0	1/1	Running	0	2m22s
> WSL:	: Ubuntu-20.04 🎖 ST-982* 😯 Python 3.8.5 64-bit ('venv': venv)	⊗ 0 ∆ ()			Ln	timer-config-jnnjj	0/1	Completed	0	2m22s
							timer-counters-0	1/1	Running	0	2m21s
							timer-timer-0	1/1	Running	0	2m21s
						8	webjivetestdevice-config-2m99t	0/1	Completed	0	2m22s
							webjivetestdevice-test-0	1/1	Running	0	2m22s
						~~~	workshopcounters-config-wltkj	0/1	Completed	0	2m22s
						- <b>5</b> 63-	workshopcounters-workshopcounters-0	1/1	Running	1	2m23s
							(venv) ubuntu@LAPTOP-5LBGJH83:~/ska-tango-e	kamples\$			



1: bash

o-examples [WSL: Ubuntu-20.04] - Visual Studio Cod



# **Development of a device step 5: tests against a real** deployment

- The make test:
  - compress the tests folder
  - create a new pod (using the image of the repository)
  - run the pytest with the true-context option
  - once done, it retrieves the files generated.
- Please note that some tests can be done only with a real deployment and therefore they are marked as "postdeployment" tests.

• Done with the "make test-deployment" target in the Makefile





∢	<u>File Edit Selection View Go Run Termina</u>	l <u>H</u> elp			output.txt - ska-tango-examp
பு	SOURCE CONTROL E @ …	M Makefile		requirements.txt	■ requirements-dev.tx
_	✓ ska-tango-exa	≡ outpu	ut.txt		
Р	Message (Ctrl+Enter to commit on 'ST-982')	1761 1762	INFO PASSE		oles.DevFactory:DevFactory:DevFactory
0	✓ Changes	1763	tests	/unit/test_tabata.p	oy::test_fatabata
င္နိုင္င်	≡output.txt ゜゜゚゚゚゚゚゚゚゚゚゚゚゚゚゚゚゚゚゚゚゚゚゚	1764			live log s
	! values.yaml charts/ska-tango-examples M	1765	INFO		oy:43 true context:
	! workshopcounters.yaml charts/ska-tan U	1766			live log ca
~	TangoWorkshopCounter.py src/ska_ta U	1767	INFO	root:test_tabat	
G	■ TangoWorkshopCounter.xmi src/ska_t U	1768 1769	INFO INFO		<pre>ples.DevFactory:DevFactory:DevFactory:DevFactory:DevFactory</pre>
<u>–</u> 0	<ul> <li>test_workshop.py tests/unit</li> <li>3, U</li> </ul>	1709	INFO		oles.DevFactory:DevFactory
	✓ .make Git % ST-982* ↔ ✓ ♡ ···	1771	INFO		ples.DevFactory:DevFactory
ß		1772	INFO		oles.DevFactory:DevF
	Message (Ctrl+Enter to commit on 'a79b3	1773	INFO		oles.DevFactory:DevFa
Д		1774	INFO		oles.DevFactory:DevFa
_		1775	PASSE	D	[ 98%]
<u>دند.</u>		1776	tests	/unit/test_workshop	<pre>p.py::test_increment</pre>
•		1777			
_		1778			======= warnings su
		1779			3.7/dist-packages/ta
		1780			3.7/dist-packages/ta
		1781			.20; for more detail
~		1782		mdArgType.DevString	g: numpy.str,
		1783 1784	Do	cs: https://docs.pv	/test.org/en/stable/
		1785			json file: /app/buil
		1786			<pre>kml file: /app/build,</pre>
		1787		_	JSON repo
		1788			reports/report.json
		1789			
		1790		coverage: pl	latform linux, pytho
		1791	Cover	age HTML written to	o dir build/htmlcov
		1792	Cover	age XML written to	<pre>file build/reports/</pre>
		1793			
		1794		• •	, 1 skipped, 1 warni
		1795	build		aut.
Q		1796		/code_analysis.stdo	but
0		1797 1798		/htmlcov/	ango examples teams
£63		1798			ango_examples_teams_ ango_examples_tabata
యో		1800		/htmlcov/status.jsd	
× w	SL: Ubuntu-20.04 🐉 ST-982* 📀 Python 3.8.5 64-b			) 1 3	

ples [W	SL: Ubuntu-20.04] - Visual	Studio Code			—
xt	! values.yaml M	🔷 test_workshop.py 3, U	≡ output.txt (Untracked) U	≣ output.txt U 🗙	! works
setup True	y.py:44 Creating P		a/1		
actor actor actor actor actor	y.py:44 Creating P y.py:44 Creating P y.py:44 Creating P y.py:44 Creating P y.py:44 Creating P y.py:44 Creating P y.py:44 Creating P	roxy for test/tabata/1 roxy for test/fatabata/1 roxy for test/counter/pre roxy for test/counter/wor roxy for test/counter/res roxy for test/counter/res	rk st cles		
: PASS	ED	[100%]			
ango/u ango/u ls and /warni ld/rep l/repo		cationWarning: `np.str` i /numpy.org/devdocs/releas	is a deprecated alias for se/1.20.0-notes.html#depre		То
on 3.7	.3-final-0				
code-	coverage.xml				
ing in	92.17s (0:01:32)	=====			
	stUpstream_py.html itpy.html				



### **Development of a device step 6: clean**

- «make uninstall-chart»
- «make reinstall-chart» if needed





Slide / 21

# **Development of a device: debugging**

- file)
- K8s allows to forward a port from the local machine to a pod (the container where the device server is running)
  - kubectl port-forward pod/tabata-tabata-0 12345:5678 -n ska-tango-examples
- Then we can make use of the vscode attach to remote process (library debugpy)

• The ska-tango-examples chart is deployed with the debugger enabled (parameter DEBUG in values.yaml





# **Debugging consideration debug_this_thread**

- debugger.
- - debugpy.debug this thread()
- https://github.com/microsoft/debugpy/wiki/API-Reference#debug_this_thread
- work on that thread.



• A standard TANGO Device server does not use Python threads so most method calls are not debuggable unless we make them aware of the

• In every method we want to debug we must add the following line of code:

 Makes the debugger aware of the current thread, and start tracing it. Must be called on any background thread that is started by means other than the usual Python APIs (i.e. the threading module), in order for breakpoints to



<u>File Edit Selection View Go Run Terminal Help</u> RUN AND DEBUG 🕨 Python: Remote At 🗸 😫 … 🛛 🌵 Tabata.py 4 🗙 ✓ VARIABLES _ska-tango-examples > src > ska_tango_examples > tabata > 🌵 Tabata.py > 😭 Tabata > 🕎 step_loop 93 self.handle_event, V Locals 94 stateless=True, > self: Tabata(test/tabata/1) 95 > Globals የያ 96 97 def step_loop(self): with tango.EnsureOmniThread(): 98 a 99 while self.get_state() == DevState.ON: with self._lock: 100 <u> </u> 101 if self.read_running_state() == RunningState.PREPARE: 102 device = self._dev_factory.get_device(self.prepCounter) 品 103 self.logger.debug("PREPARE %s", device.value) 104 device.decrement() 105 if self.read_running_state() == RunningState.WORK: Д 106 device = self._dev_factory.get_device(self.workCounter) self.logger.debug("WORK %s", device.value) 107 device.decrement() ✓ WATCH 108 e: NameError: name 'e' is not defined 109 if self.read_running_state() == RunningState.REST: device = self._dev_factory.get_device(self.restCounter) 110 9 111 self.logger.debug("REST %s", device.value) 112 device.decrement() ۲ OUTPUT DEBUG CONSOLE TERMINAL PROBLEMS 4 Active Status: No resource quota. No LimitRange resource. install-chart: install charts/test-parent/ release: test in Namespace: ska-tango-examples with params: -set global.minikube=true --set global.tango_host=tango-host-databaseds-from-makefile-test:10000 --set ska -tango-base.display=172.25.16.1:0.0 --set ska-tango-base.xauthority=/home/ubuntu/.Xauthority --set ska-ta ngo-base.jive.enabled=true --set webjive.enabled=false --set tango_example.tango_example.image.tag=0.4.15 --set event_generator.events_generator.image.tag=0.4.15 --values gilab_values.yaml ✓ CALL STACK helm upgrade --install test \ > MainThread PAUSED --set global.minikube=true --set global.tango_host=tango-host-databaseds-from-makefile-test:10000 --set s > Dummy-7 PAUSED ka-tango-base.display=172.25.16.1:0.0 --set ska-tango-base.xauthority=/home/ubuntu/.Xauthority --set ska-✓ Thread-8 PAUSED ON BREAKPOINT tango-base.jive.enabled=true --set webjive.enabled=false --set tango_example.tango_example.image.tag=0.4. 15 --set event_generator.events_generator.image.tag=0.4.15 --values gilab_values.yaml \ Tabata.py 99:1 step_loop charts/test-parent/ --namespace ska-tango-examples; \ threading.py 865:1 run rm gilab values.yaml Release "test" does not exist. Installing it now. ✓ BREAKPOINTS NAME: test Raised Exceptions LAST DEPLOYED: Wed Oct 13 15:02:04 2021 Uncaught Exceptions NAMESPACE: ska-tango-examples STATUS: deployed

User Uncaught Exceptions

abstract_command.py ska-tmc-centralno... 258 command_executor.py ska-tmc-centralno... 94 command_executor.py ska-tmc-centralno... 96 V helper_adapter_factory.py ska-tmc-central... 24

Image: Sector State S

helper_adapter_factory.py ska-tmc-central... 30

× WSL: Ubuntu-20.04 & TANGO-workshop ↔ Python 3.8.5 64-bit ('venv': venv) ⊗ 0 🛆 4 🕹 Python: Remote Attach (ska-tango-examples)

REVISION: 1

ango-examples

TEST SUITE: None

Forwarding from 127.0.0.1:12345 -> 5678

Forwarding from [::1]:12345 -> 5678

Handling connection for 12345

(R)

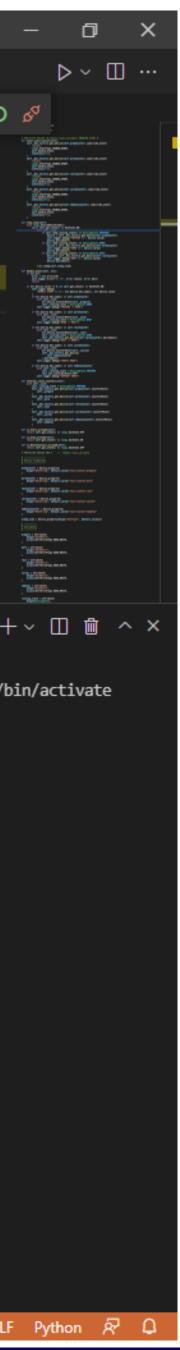
鎹

source /home/ubuntu/ska-tango-examples/venv/bin/activate ubuntu@LAPTOP-5LBGJH83:~/ska-tango-examples\$ source /home/ubuntu/ska-tango-examples/venv/bin/activate (venv) ubuntu@LAPTOP-5LBGJH83:~/ska-tango-examples\$

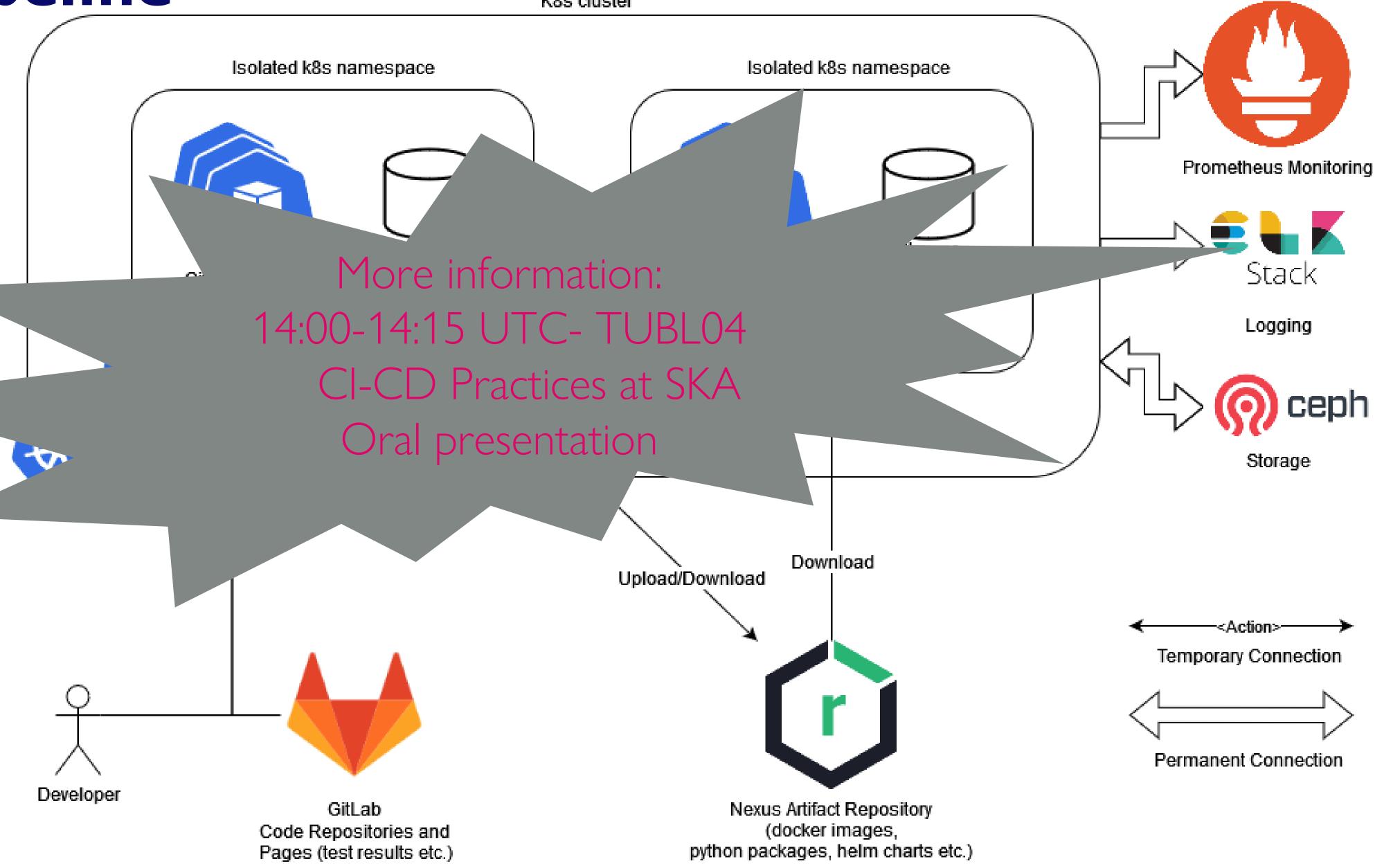
ubuntu@LAPTOP-5LBGJH83:~/ska-tango-examples\$ kubectl port-forward pod/tabata-tabata-0 12345:5678 -n ska-t

1: kubectl (ska-tango-e: 🗸 🕂 🗸 🔟 🏛 \land 🗙

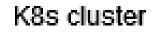
#### ା 🗅 🖓 🏌 🏌 ପ 🖋



# **Gitlab pipeline**











#### Thanks for your attention

We recognise and acknowledge the Indigenous peoples and cultures that have traditionally lived on the lands on which our facilities are located.







•

 $\bullet$ 

•