[Where is the complete list of kubernetes objects?](https://stackoverflow.com/questions/53053888/where-is-the-complete-list-of-kubernetes-objects)

17

kubernetes seems to have lot of objects. I can't seem to find the full list of objects anywhere. After briefly searching on google, I can find results which mention a subset of kubernetes objects. Is the ***full*** list of objects documented somewhere, perhaps in source code? Thank you.

Following command successfully display all kubernetes objects

**$ kubectl api-resources**

**Example**

[root@hsk-controller ~]# kubectl api-resources

NAME SHORTNAMES KIND

bindings Binding

componentstatuses cs ComponentStatus

configmaps cm ConfigMap

endpoints ep Endpoints

events ev Event

limitranges limits LimitRange

namespaces ns Namespace

nodes no Node

persistentvolumeclaims pvc PersistentVolumeClaim

persistentvolumes pv PersistentVolume

pods po Pod

podtemplates PodTemplate

replicationcontrollers rc ReplicationController

resourcequotas quota ResourceQuota

secrets Secret

serviceaccounts sa ServiceAccount

services svc Service

initializerconfigurations InitializerConfiguration

mutatingwebhookconfigurations MutatingWebhookConfiguration

validatingwebhookconfigurations ValidatingWebhookConfiguration

customresourcedefinitions crd,crds CustomResourceDefinition

apiservices APIService

controllerrevisions ControllerRevision

daemonsets ds DaemonSet

deployments deploy Deployment

replicasets rs ReplicaSet

statefulsets sts StatefulSet

tokenreviews TokenReview

localsubjectaccessreviews LocalSubjectAccessReview

selfsubjectaccessreviews SelfSubjectAccessReview

selfsubjectrulesreviews SelfSubjectRulesReview

subjectaccessreviews SubjectAccessReview

horizontalpodautoscalers hpa HorizontalPodAutoscaler

cronjobs cj CronJob

jobs Job

brpolices br,bp BrPolicy

clusters rcc Cluster

filesystems rcfs Filesystem

objectstores rco ObjectStore

pools rcp Pool

certificatesigningrequests csr CertificateSigningRequest

leases Lease

events ev Event

daemonsets ds DaemonSet

deployments deploy Deployment

ingresses ing Ingress

networkpolicies netpol NetworkPolicy

podsecuritypolicies psp PodSecurityPolicy

replicasets rs ReplicaSet

nodes NodeMetrics

pods PodMetrics

networkpolicies netpol NetworkPolicy

poddisruptionbudgets pdb PodDisruptionBudget

podsecuritypolicies psp PodSecurityPolicy

clusterrolebindings ClusterRoleBinding

clusterroles ClusterRole

rolebindings RoleBinding

roles Role

volumes rv Volume

priorityclasses pc PriorityClass

storageclasses sc StorageClass

volumeattachments VolumeAttachment

**Note**: kubernate version is v1.12\*

kubectl version

The following command list all supported API versions:

**$ kubectl api-versions**

You can have a bit detailed information from kube-apiserver REST API:

Open connection to kube-apiserver

$ kubectl proxy &

Now you can discover API resources:

This request gives you all existed paths on apiserver (in JSON format):

$ curl http://localhost:8001/

"/apis/extensions/v1beta1",

"/apis/networking.k8s.io",

"/apis/networking.k8s.io/v1",

"/apis/policy",

"/apis/policy/v1beta1",

"/apis/rbac.authorization.k8s.io",

"/apis/rbac.authorization.k8s.io/v1",

...

"/version"

]

}

You can request details about particular path:

curl http://localhost:8001/api/v1

...

{

"name": "configmaps",

"singularName": "",

"namespaced": true,

"kind": "ConfigMap",

"verbs": [

"create",

"delete",

"deletecollection",

"get",

"list",

"patch",

"update",

"watch"

],

"shortNames": [

"cm"

]

},

...

This information helps you to write kubectl commands, e.g.:

$ kubectl get configmaps

$ kubectl get cm

But you may find more convenient to use built-in documentation provided by kubectl explain.

For example, this command shows you a list of Kubernetes objects:

$ kubectl explain

You can have detailed information about any of listed resources:

**$ kubectl explain rc**

**$ kubectl explain rc.spec**

**$ kubectl explain rc.spec.selector**

Or you can print full blown YAML template(or part) of the object by adding --recursive flag:

**$ kubectl explain rc --recursive**

**$ kubectl explain rc.metadata –recursive**

Links in the desctiption points to the documentation about particular object. E.g.:

DESCRIPTION:

If the Labels of a ReplicationController are empty, they are defaulted to

be the same as the Pod(s) that the replication controller manages. Standard

object's metadata. More info:

https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata

ObjectMeta is metadata that all persisted resources must have, which

includes all objects users must create.

If you need complete description with examples you can always find it in the official [API Reference](https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.12/) (or the [older version](https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.10/)), mentioned by [Matthew L Daniel](https://stackoverflow.com/users/225016/matthew-l-daniel)

You also might find helpful [kubectl Reference](https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands) or [kubectl Cheatsheet](https://kubernetes.io/docs/reference/kubectl/cheatsheet/)

**Update:** Using the following one-liner you can list all objects grouped by API versions (including CRDs). It may be useful to check if an object is present in more than one API group and therefore more than one apiVersion is applicable in its manifest. (For different apiVersions object configuration may be slightly different.)

**$ a=$(kubectl api-versions) ; for n in $a ; do echo ; echo "apiVersion: $n" ;**

**kubectl api-resources --api-group="${n%/\*}" ; done**

Partial example output:

...

apiVersion: autoscaling/v1

NAME SHORTNAMES APIGROUP NAMESPACED KIND

horizontalpodautoscalers hpa autoscaling true HorizontalPodAutoscaler

apiVersion: autoscaling/v2beta1

NAME SHORTNAMES APIGROUP NAMESPACED KIND

horizontalpodautoscalers hpa autoscaling true HorizontalPodAutoscaler

apiVersion: autoscaling/v2beta2

NAME SHORTNAMES APIGROUP NAMESPACED KIND

horizontalpodautoscalers hpa autoscaling true HorizontalPodAutoscaler

apiVersion: batch/v1

NAME SHORTNAMES APIGROUP NAMESPACED KIND

cronjobs cj batch true CronJob

jobs batch true Job

apiVersion: batch/v1beta1

NAME SHORTNAMES APIGROUP NAMESPACED KIND

cronjobs cj batch true CronJob

jobs batch true Job

Great answer @VAS.

**$ kubectl api-resources | cut -c92-150**

gives me all the kubernetes object types. Found 51 types.

–

[user674669](https://stackoverflow.com/users/674669/user674669" \o "8,594 reputation)

[Oct 30, 2018 at 21:06](https://stackoverflow.com/questions/53053888/where-is-the-complete-list-of-kubernetes-objects#comment93044236_53066562)

2

You’re right. Starting from Jul 17 (PR #9437) command ‘kubectl api-resources’ was introduced. Behavior of ‘kubectl explain’ has also changed. I’ve used older version of Kubernetes cluster to prepare the answer.

–

**web** <https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.16/>

**resources list**

**$ kubectl api-resources -o wide**

NAME SHORTNAMES APIGROUP NAMESPACED KIND VERBS

bindings true Binding [create]

componentstatuses cs false ComponentStatus [get list]

configmaps cm true ConfigMap [create delete deletecollection get list patch update watch]

endpoints ep true Endpoints [create delete deletecollection get list patch update watch]

events ev true Event [create delete deletecollection get list patch update watch]

limitranges limits true LimitRange [create delete deletecollection get list patch update watch]

namespaces ns false Namespace [create delete get list patch update watch]

nodes no false Node [create delete deletecollection get list patch update watch]

persistentvolumeclaims pvc true PersistentVolumeClaim [create delete deletecollection get list patch update watch]

persistentvolumes pv false PersistentVolume [create delete deletecollection get list patch update watch]

pods po true Pod [create delete deletecollection get list patch update watch]

podtemplates true PodTemplate [create delete deletecollection get list patch update watch]

replicationcontrollers rc true ReplicationController [create delete deletecollection get list patch update watch]

resourcequotas quota true ResourceQuota [create delete deletecollection get list patch update watch]

secrets true Secret [create delete deletecollection get list patch update watch]

serviceaccounts sa true ServiceAccount [create delete deletecollection get list patch update watch]

services svc true Service [create delete get list patch update watch]

mutatingwebhookconfigurations admissionregistration.k8s.io false MutatingWebhookConfiguration [create delete deletecollection get list patch update watch]

validatingwebhookconfigurations admissionregistration.k8s.io false ValidatingWebhookConfiguration [create delete deletecollection get list patch update watch]

customresourcedefinitions crd,crds apiextensions.k8s.io false CustomResourceDefinition [create delete deletecollection get list patch update watch]

apiservices apiregistration.k8s.io false APIService [create delete deletecollection get list patch update watch]

controllerrevisions apps true ControllerRevision [create delete deletecollection get list patch update watch]

daemonsets ds apps true DaemonSet [create delete deletecollection get list patch update watch]

deployments deploy apps true Deployment [create delete deletecollection get list patch update watch]

replicasets rs apps true ReplicaSet [create delete deletecollection get list patch update watch]

statefulsets sts apps true StatefulSet [create delete deletecollection get list patch update watch]

tokenreviews authentication.k8s.io false TokenReview [create]

localsubjectaccessreviews authorization.k8s.io true LocalSubjectAccessReview [create]

selfsubjectaccessreviews authorization.k8s.io false SelfSubjectAccessReview [create]

selfsubjectrulesreviews authorization.k8s.io false SelfSubjectRulesReview [create]

subjectaccessreviews authorization.k8s.io false SubjectAccessReview [create]

horizontalpodautoscalers hpa autoscaling true HorizontalPodAutoscaler [create delete deletecollection get list patch update watch]

cronjobs cj batch true CronJob [create delete deletecollection get list patch update watch]

jobs batch true Job [create delete deletecollection get list patch update watch]

certificatesigningrequests csr certificates.k8s.io false CertificateSigningRequest [create delete deletecollection get list patch update watch]

leases coordination.k8s.io true Lease [create delete deletecollection get list patch update watch]

events ev events.k8s.io true Event [create delete deletecollection get list patch update watch]

ingresses ing extensions true Ingress [create delete deletecollection get list patch update watch]

ingresses ing networking.k8s.io true Ingress [create delete deletecollection get list patch update watch]

networkpolicies netpol networking.k8s.io true NetworkPolicy [create delete deletecollection get list patch update watch]

runtimeclasses node.k8s.io false RuntimeClass [create delete deletecollection get list patch update watch]

poddisruptionbudgets pdb policy true PodDisruptionBudget [create delete deletecollection get list patch update watch]

podsecuritypolicies psp policy false PodSecurityPolicy [create delete deletecollection get list patch update watch]

clusterrolebindings rbac.authorization.k8s.io false ClusterRoleBinding [create delete deletecollection get list patch update watch]

clusterroles rbac.authorization.k8s.io false ClusterRole [create delete deletecollection get list patch update watch]

rolebindings rbac.authorization.k8s.io true RoleBinding [create delete deletecollection get list patch update watch]

roles rbac.authorization.k8s.io true Role [create delete deletecollection get list patch update watch]

priorityclasses pc scheduling.k8s.io false PriorityClass [create delete deletecollection get list patch update watch]

csidrivers storage.k8s.io false CSIDriver [create delete deletecollection get list patch update watch]

csinodes storage.k8s.io false CSINode [create delete deletecollection get list patch update watch]

storageclasses sc storage.k8s.io false StorageClass [create delete deletecollection get list patch update watch]

volumeattachments storage.k8s.io false VolumeAttachment [create delete deletecollection get list patch update watch]

**details about each object**

**$ kubectl explain --help**

List the fields for supported resources

This command describes the fields associated with each supported API resource. Fields are identified via a simple JSONPath identifier:

<type>.<fieldName>[.<fieldName>]

Add the --recursive flag to display all of the fields at once without descriptions. Information about each field is retrieved from the server in OpenAPI format.

Use "kubectl api-resources" for a complete list of supported resources.

Examples:

# Get the documentation of the resource and its fields

**$ kubectl explain pods**

# Get the documentation of a specific field of a resource

**$ kubectl explain pods.spec.containers**

Options:

--api-version='': Get different explanations for particular API version

--recursive=false: Print the fields of fields (Currently only 1 level deep)

Usage:

kubectl explain RESOURCE [options]

Use "kubectl options" for a list of global command-line options (applies to all commands).

I've been frustrated by the same issue. While you've got some good answers, I wanted something that 1) Was grouped by api version 2) was just a list of names instead of a book of documentation. I've been sorting out our RBAC, and it's a bit tricky without that. Couldn't find one, so here's the one I made (v1.18.0):

v1

bindings

componentstatuses

configmaps

endpoints

events

limitranges

namespaces

namespaces/finalize

namespaces/status

nodes

nodes/proxy

nodes/status

persistentvolumeclaims

persistentvolumeclaims/status

persistentvolumes

persistentvolumes/status

pods

pods/attach

pods/binding

pods/eviction

pods/exec

pods/log

pods/portforward

pods/proxy

pods/status

podtemplates

replicationcontrollers

replicationcontrollers/scale

replicationcontrollers/status

resourcequotas

resourcequotas/status

secrets

serviceaccounts

serviceaccounts/token

services

services/proxy

services/status

admissionregistration.k8s.io/v1

mutatingwebhookconfigurations

validatingwebhookconfigurations

admissionregistration.k8s.io/v1beta1

mutatingwebhookconfigurations

validatingwebhookconfigurations

apiextensions.k8s.io/v1

customresourcedefinitions

customresourcedefinitions/status

apiextensions.k8s.io/v1beta1

customresourcedefinitions

customresourcedefinitions/status

apiregistration.k8s.io/v1

apiservices

apiservices/status

apiregistration.k8s.io/v1beta1

apiservices

apiservices/status

apps/v1

controllerrevisions

daemonsets

daemonsets/status

deployments

deployments/scale

deployments/status

replicasets

replicasets/scale

replicasets/status

statefulsets

statefulsets/scale

statefulsets/status

authentication.k8s.io/v1

tokenreviews

authentication.k8s.io/v1beta1

tokenreviews

authorization.k8s.io/v1

localsubjectaccessreviews

selfsubjectaccessreviews

selfsubjectrulesreviews

subjectaccessreviews

authorization.k8s.io/v1beta1

localsubjectaccessreviews

selfsubjectaccessreviews

selfsubjectrulesreviews

subjectaccessreviews

autoscaling/v1

horizontalpodautoscalers

horizontalpodautoscalers/status

autoscaling/v2beta1

horizontalpodautoscalers

horizontalpodautoscalers/status

autoscaling/v2beta2

horizontalpodautoscalers

horizontalpodautoscalers/status

batch/v1

jobs

jobs/status

batch/v1beta1

cronjobs

cronjobs/status

certificates.k8s.io/v1beta1

certificatesigningrequests

certificatesigningrequests/approval

certificatesigningrequests/status

coordination.k8s.io/v1

leases

coordination.k8s.io/v1beta1

leases

crd.k8s.amazonaws.com/v1alpha1

eniconfigs

events.k8s.io/v1beta1

events

extensions/v1beta1

ingresses

ingresses/status

metrics.k8s.io/v1beta1

nodes

pods

networking.k8s.io/v1

networkpolicies

networking.k8s.io/v1beta1

ingresses

ingresses/status

node.k8s.io/v1beta1

runtimeclasses

policy/v1beta1

poddisruptionbudgets

poddisruptionbudgets/status

podsecuritypolicies

rbac.authorization.k8s.io/v1

clusterrolebindings

clusterroles

rolebindings

roles

rbac.authorization.k8s.io/v1beta1

clusterrolebindings

clusterroles

rolebindings

roles

scheduling.k8s.io/v1

priorityclasses

scheduling.k8s.io/v1beta1

priorityclasses

storage.k8s.io/v1

storageclasses

volumeattachments

volumeattachments/status

storage.k8s.io/v1beta1

csidrivers

csinodes

storageclasses

volumeattachments

----------------------------------------------------------------------------------

# install crane

-----------------------------------------------------------------------------------

# download the crane releases

$ curl -sL https://api.github.com/repos/konveyor/crane/releases/latest | jq -r ".assets[] | select(.name | contains(\"amd64-linux\")) | .browser\_download\_url" | wget -i-

# rename the file to crane

atlantis@kubuntu:~/oc$ mv amd64-linux-crane-v0.0.3 crane

$ chmod +x ./crane

# set KUBECONFIG env variable

$ export KUBECONFIG=/home/atlantis/.kube/config

# ensure the cluster is bound to the correct current-context

$ oc config current-context

# call the crane to lift the resources from the given cluster and the given namespace

$ crane export -n myproject -e minishift-demo-project --context myproject/192-168-99-101:8443/admin

---------------------------------------------------------------------------------------------------------------------------------------------------------------------

DOCKER AND DOCKER COMPOSE

----------------------------------------------------------------------------------------------------------------------------------------------------------------------

# install docker

sudo apt install docker

sudo apt install docker.io

# install docker compose

sudo apt install docker-compose

----------------------------------------------------------------------------------------------------------------------------------------------------------------------

SHIFTER COMPILING AND TESTING

----------------------------------------------------------------------------------------------------------------------------------------------------------------------

# install go in

/home/atlantis/go

export set PATH:$PATH:/home/atlantis/go/bin

# create a directory to host the shifter git repo

mkdir /home/atlantis/scripts/shifter

cd /home/atlantis/scripts/shifter

git clone https://github.com/google/shifter.git

cd /home/atlantis/scripts/shifter/shifter/shifter

# run shifter with the go interpreter

$ make

or

$ go run shifter.go

# build an executable

$ go build -o bin/shifter shifter.go

# copy the shifter executable to the oc folder where we have the oc and kubectl binary

$ cp ./bin/shifter /home/atlantis/oc/shifter

# test

atlantis@kubuntu:~/scripts/shifter/shifter/shifter> shifter convert -t yaml -f ./\_test/yaml/multidoc/os-nginx.yaml -o ./melvin -i yaml

# verify the generated yaml files

ls -la ./melvin/

go run . convert -t yaml -f ./\_test/yaml/multidoc/os-nginx.yaml -o ./out -i yaml

2022/06/01 13:04:55

\_\_\_\_\_ \_\_ \_ \_\_\_\_\_\_

/ \_\_\_// /\_ (\_) \_\_/ /\_\_\_\_ \_\_\_\_\_

\\_\_ \/ \_\_ \/ / /\_/ \_\_/ \_ \/ \_\_\_/

\_\_\_/ / / / / / \_\_/ /\_/ \_\_/ /

/\_\_\_\_/\_/ /\_/\_/\_/ \\_\_/\\_\_\_/\_/

----------------------------------------

2022/06/01 13:04:55 Converting yaml ./\_test/yaml/multidoc/os-nginx.yaml to yaml ./out

2022/06/01 13:04:55 Generating file 0 -

2022/06/01 13:04:55 Generating file 1 -

2022/06/01 13:04:55 Generating file 2 -

2022/06/01 13:04:55 Generating file 3 -

2022/06/01 13:04:55 Generating file 4 -

go run . convert -t yaml -f ./\_test/yaml/multifile/ -o ./out/files -i yaml

2022/06/01 13:04:57

\_\_\_\_\_ \_\_ \_ \_\_\_\_\_\_

/ \_\_\_// /\_ (\_) \_\_/ /\_\_\_\_ \_\_\_\_\_

\\_\_ \/ \_\_ \/ / /\_/ \_\_/ \_ \/ \_\_\_/

\_\_\_/ / / / / / \_\_/ /\_/ \_\_/ /

/\_\_\_\_/\_/ /\_/\_/\_/ \\_\_/\\_\_\_/\_/

----------------------------------------

2022/06/01 13:04:57 Converting yaml ./\_test/yaml/multifile/ to yaml ./out/files

2022/06/01 13:04:57 Generating file 0 -

2022/06/01 13:04:57 Generating file 1 -

2022/06/01 13:04:57 Generating file 2 -

2022/06/01 13:04:57 Generating file 3 -

2022/06/01 13:04:57 Generating file 0 -

2022/06/01 13:04:57 Generating file 0 -

2022/06/01 13:04:57 Generating file 1 -

2022/06/01 13:04:57 Generating file 0 -

2022/06/01 13:04:57 Generating file 0 -

go run . convert -t yaml -i yaml -f ./\_test/yaml/deploymentconfig.yaml -o ./out/dc

2022/06/01 13:04:59

\_\_\_\_\_ \_\_ \_ \_\_\_\_\_\_

/ \_\_\_// /\_ (\_) \_\_/ /\_\_\_\_ \_\_\_\_\_

\\_\_ \/ \_\_ \/ / /\_/ \_\_/ \_ \/ \_\_\_/

\_\_\_/ / / / / / \_\_/ /\_/ \_\_/ /

/\_\_\_\_/\_/ /\_/\_/\_/ \\_\_/\\_\_\_/\_/

----------------------------------------

2022/06/01 13:04:59 Converting yaml ./\_test/yaml/deploymentconfig.yaml to yaml ./out/dc

2022/06/01 13:04:59 Generating file 0 -

go run . convert -t yaml -f ./\_test/yaml/multifile/ -o ./out/files.yaml -i yaml

2022/06/01 13:05:00

\_\_\_\_\_ \_\_ \_ \_\_\_\_\_\_

/ \_\_\_// /\_ (\_) \_\_/ /\_\_\_\_ \_\_\_\_\_

\\_\_ \/ \_\_ \/ / /\_/ \_\_/ \_ \/ \_\_\_/

\_\_\_/ / / / / / \_\_/ /\_/ \_\_/ /

/\_\_\_\_/\_/ /\_/\_/\_/ \\_\_/\\_\_\_/\_/

----------------------------------------

2022/06/01 13:05:00 Converting yaml ./\_test/yaml/multifile/ to yaml ./out/files.yaml

2022/06/01 13:05:00 Generating file 0 -

2022/06/01 13:05:00 Generating file 1 -

2022/06/01 13:05:00 Generating file 2 -

2022/06/01 13:05:00 Generating file 3 -

2022/06/01 13:05:01 Generating file 0 -

2022/06/01 13:05:01 Generating file 0 -

2022/06/01 13:05:01 Generating file 1 -

2022/06/01 13:05:01 Generating file 0 -

2022/06/01 13:05:01 Generating file 0 -

go run . convert -t yaml -i yaml -f ./\_test/yaml/quoted\_nested\_strings.yaml -o ./out/quoted\_nested\_strings.yaml

2022/06/01 13:05:02

\_\_\_\_\_ \_\_ \_ \_\_\_\_\_\_

/ \_\_\_// /\_ (\_) \_\_/ /\_\_\_\_ \_\_\_\_\_

\\_\_ \/ \_\_ \/ / /\_/ \_\_/ \_ \/ \_\_\_/

\_\_\_/ / / / / / \_\_/ /\_/ \_\_/ /

/\_\_\_\_/\_/ /\_/\_/\_/ \\_\_/\\_\_\_/\_/

----------------------------------------

2022/06/01 13:05:02 Converting yaml ./\_test/yaml/quoted\_nested\_strings.yaml to yaml ./out/quoted\_nested\_strings.yaml

2022/06/01 13:05:02 Generating file 0 -

go run . convert -t helm -i template -f ./\_test/os-nginx-template.yaml -o ./out/helm

2022/06/01 13:05:04

\_\_\_\_\_ \_\_ \_ \_\_\_\_\_\_

/ \_\_\_// /\_ (\_) \_\_/ /\_\_\_\_ \_\_\_\_\_

\\_\_ \/ \_\_ \/ / /\_/ \_\_/ \_ \/ \_\_\_/

\_\_\_/ / / / / / \_\_/ /\_/ \_\_/ /

/\_\_\_\_/\_/ /\_/\_/\_/ \\_\_/\\_\_\_/\_/

----------------------------------------

2022/06/01 13:05:04 Converting template ./\_test/os-nginx-template.yaml to helm ./out/helm

2022/06/01 13:05:04 Converting object Route

2022/06/01 13:05:04 Converting object DeploymentConfig

2022/06/01 13:05:04 Converting object Service

2022/06/01 13:05:04 Converting object ImageStream

2022/06/01 13:05:04 Converting object PersistentVolumeClaim

2022/06/01 13:05:04 Converting object ConfigMap

2022/06/01 13:05:04 Converting object ConfigMap

2022/06/01 13:05:04 Writing helm template file 0 Ingress

2022/06/01 13:05:04 Writing helm template file 1 Deployment

2022/06/01 13:05:04 Writing helm template file 2 Service

2022/06/01 13:05:04 Writing helm template file 3 PersistentVolumeClaim

2022/06/01 13:05:04 Writing helm template file 4 ConfigMap

2022/06/01 13:05:04 Writing helm template file 5 ConfigMap

-----------------------------------------------------------------------------------

SHIFTER SERVER AND UI

-----------------------------------------------------------------------------------

atlantis@kubuntu:~/scripts/shifter/shifter/shifter$ docker-compose -f ./docker-compose.yml up

-----------------------------------------------------------------------------------