Using k8clone to migrate application between k8s clusters.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Version | Date | Description | Modified By | Document Name |
| 1 | V0.1 | Sep 21 2023 | Draft version | Hugo Juárez | Using k8clone to migrate application between k8s clusters. |
| 2 | V0.2 | Sep 27 2023 | Second Revision | Hugo Juárez | Using k8clone to migrate application between k8s clusters. |

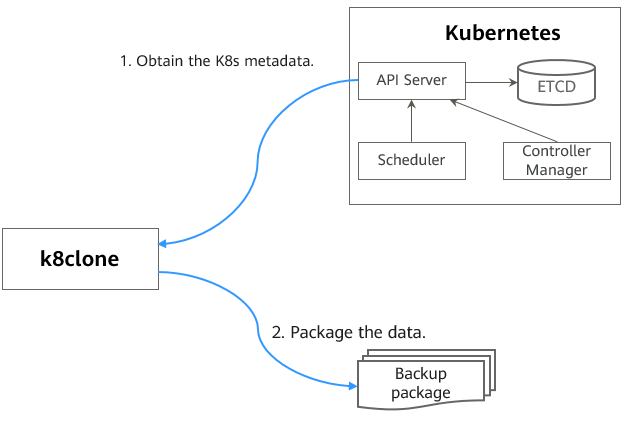
# Background:

When performing application migration from clusters in an on-premises IDC is comprised of two steps, application backup and application migration. In simpler terms, the application is first backed up and then migrated to the destination cluster using data restoration.

# Summary of tools used:

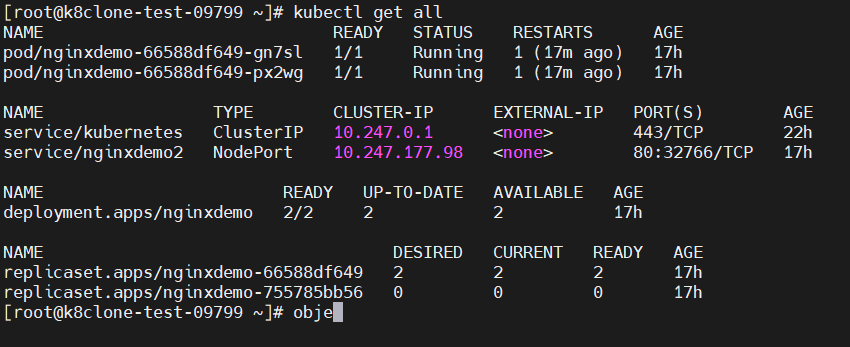
K8CLONE is a simple Kubernetes metadata cloning tool. It can save Kubernetes metadata (objects) as a local package and restore the metadata to the destination cluster. In this way, applications can be migrated from clusters in an on-premises IDC to the cloud.

K8CLONE can run on Linux (x86 and ARM) and Windows. The usage is similar in both environments.

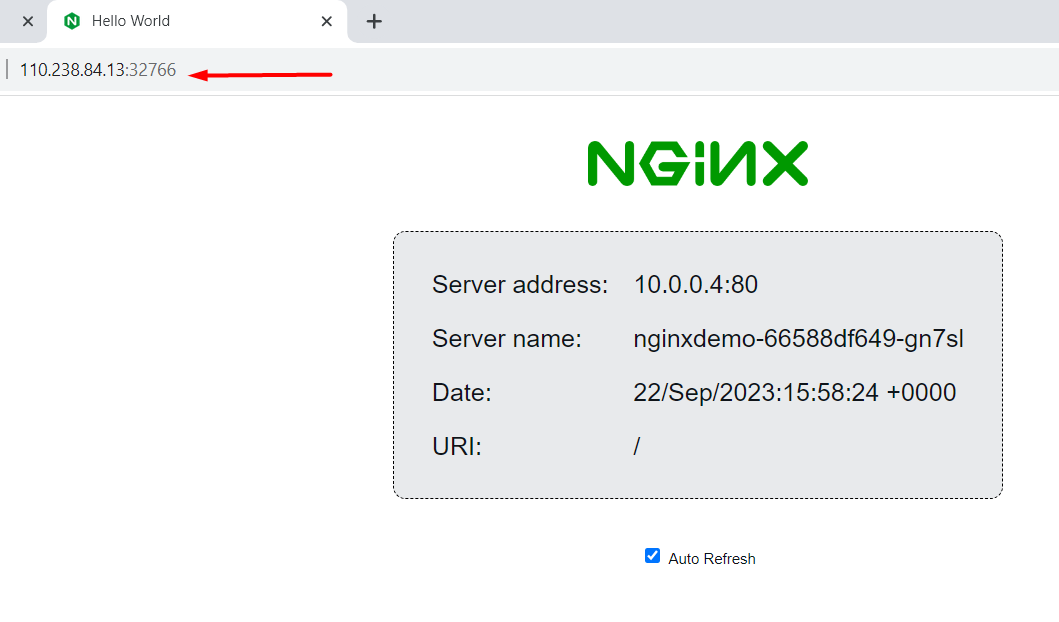
Data backup process

# Example Guide

In this example guide, an application from a kubernetes cluster (**k8clone-test)** will be migrated to a new cluster named k8clone-destination. Using kubectl **get all** we can see a list of all pods, services, daemon sets, deployments, replica sets, jobs, cronjobs, and stateful sets.



Currently the cluster has 1 deployment with two pods of the nginxdemo. You can use a local browser to see the application running the demo.



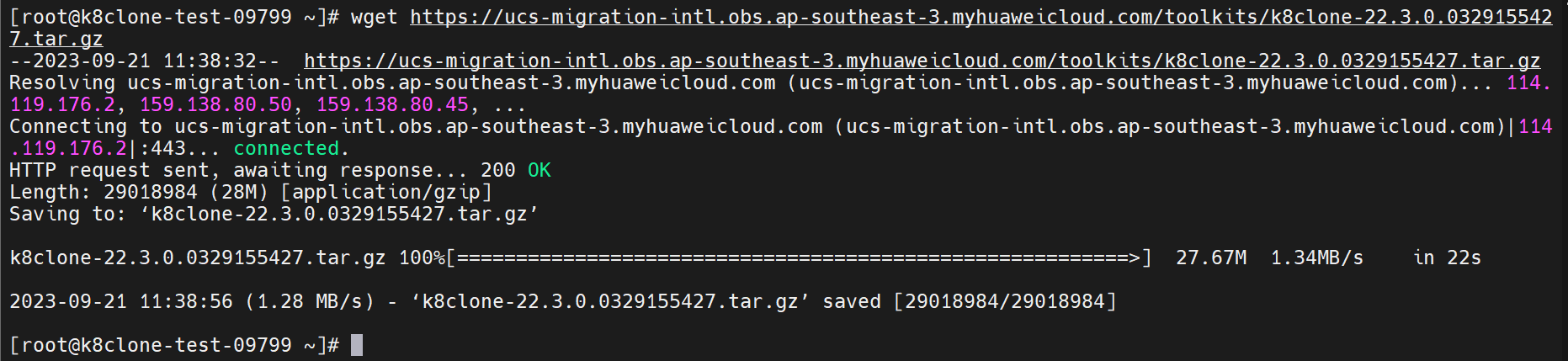
Considerations:

Ensure that services on which the cloud native application has dependency on have already been migrated. Also perform data back up during off-peak hours.

In this example the Linux x86 version of k8clone is going to be used.

Backing up data from a cluster:

Step 1. Download the tool from: <https://ucs-migration-intl.obs.ap-southeast-3.myhuaweicloud.com/toolkits/k8clone-22.3.0.0329155427.tar.gz>



Step 2. Extract the file with:

$ tar -xvf [**k8clone-22.3.0.0329155427.tar.gz]**

Step 3. Run chmod to grant execute permission to each tool.

$ chmod u+x **tool\_name**

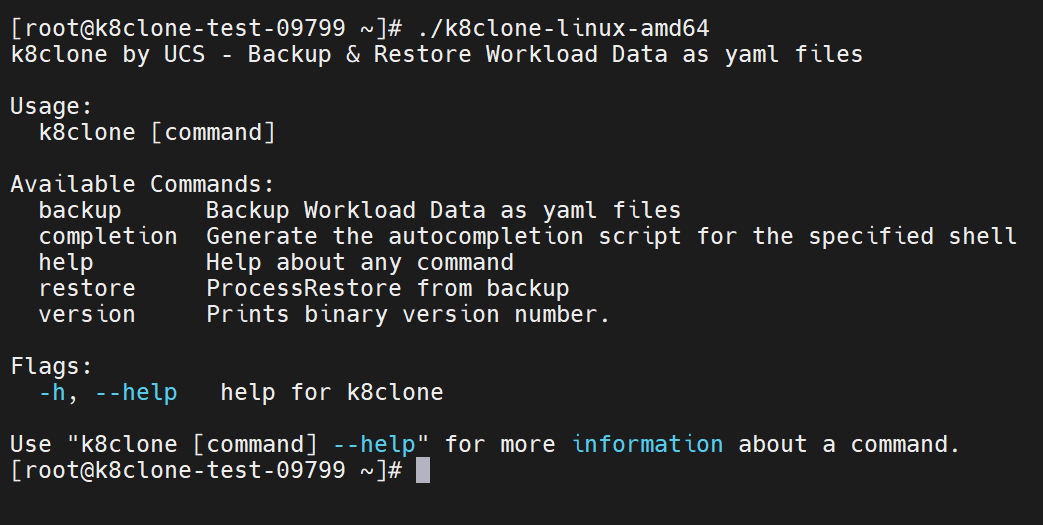
Step 4. Connect to the source cluster using kubectl. For details go to: [Connecting to a Cluster Using kubectl](https://support.huaweicloud.com/intl/en-us/usermanual-cce/cce_10_0107.html#section2).

Step 4. Run the tool.

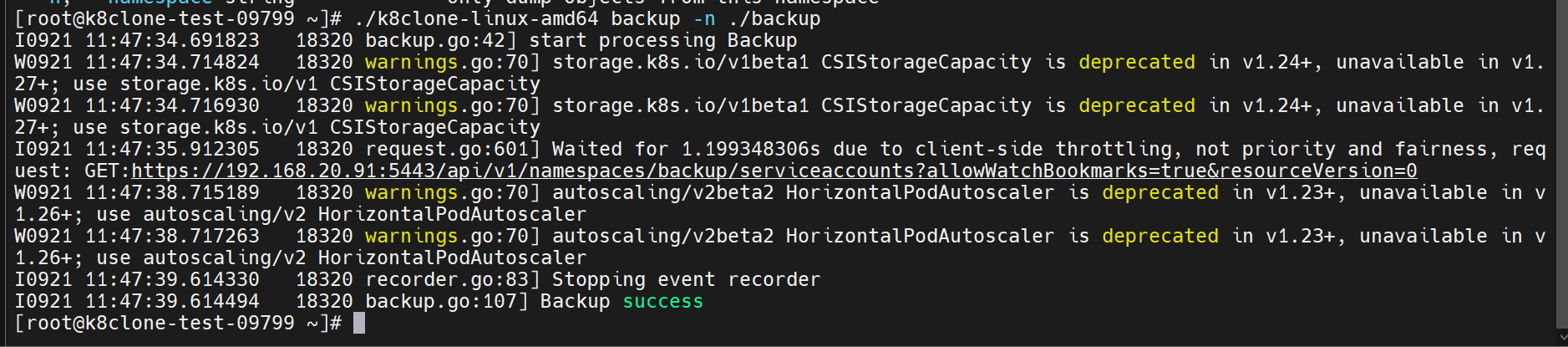
**Windows**: run using k8clone-windows-amd64.exe

**Linux**: run ./k8clone-linux-amd64

Running the tool without any parameters will show the available commands.



Step 5. Use the backup command to run the tool, the default output path will be the same folder where the tool is located.

$ k8clone backup 

On completion k8clone generates the file **k8clone-dump.zip**

Step 6. Copy the **k8clone-dump.zip file** to the destination cluster.

Restoring data using k8clone.

Before restoration, a data restoration configuration file (**restore.json**) needs to be prepared first. This file specifies the Storage Class, and Image Repositories on the destination Cluster.

**Preparing the file restore.json.**

The data restoration configuration file “***restore.json***” allows to automatically change the storage class names of PVC and StatefulSet along with the repository address of the image used by the workload during application restoration.

The format of the file is the following:

An example of the file would look like this:

{

“StorageClass”:{

“csi-disk” : “csi-disk-new”

},

“ImageRepo”:{

“quay.io/coreos” : “swr.cn-north-4.myhuaweicloud.com/paas”

}

}

{

“StorageClass”:

“OldStorageClassName” : “NewStorageClassName” //The StorageClassName filed of PVC and StatefulSet can be changed.

“ImageRepo”:

“OldImageRepo1” : “NewImageRepo1” //eg: “dockerhub.com”:”cn-north-4.swr.huaweicloud.com”

“OldImageRepo2” : “NewImageRepo2” //eg: “dockerhub.com/org1”:”cn-north-4.swr.huaweicloud.com/org2”

“NoRepo” : “NewImageRepo3” //eg:”golang”:”swr.cn-north-4.myhuaweicloud.com/paas/golang”

}

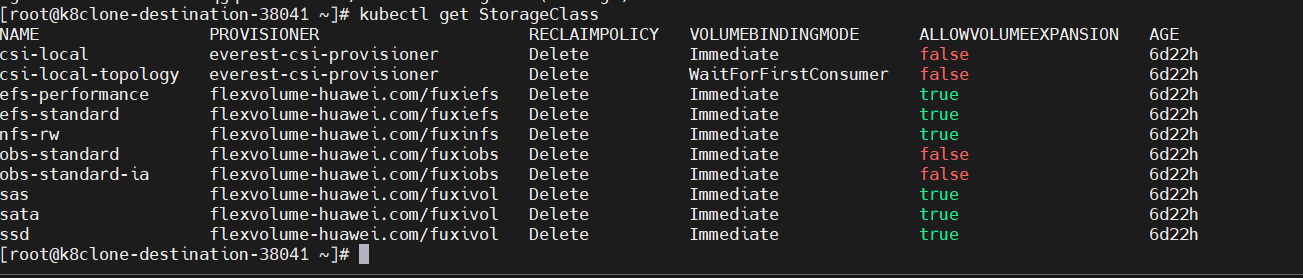
**StorageClass:** The storage class names of PVC and VolumeClaimTemplates can be automatically changed based on settings.

**ImageRepo:** The repository address of the image used by the workload can be changed. The workload can be Deployment (including initContainer). StatefulSet, Orphaned Pod, Job, CronJob, Replica Set, Replication Controller and DaemonSet.

Step 1. Query the storage class name from your on-premises k8s cluster.

You can use the following command to get the storage class for the selected namespace you are going to backup.

$ kubectl get StorageClass



You can use this command to get the storage class for both the source and destination cluster.

Step 2. Specify the new repository for the images in your k8s cluster. That would be your SWR repository if needed.

Step 3. Change the restore.json file as needed.

**Restoring the data.**

Step 1. Connect to the Destination Cluster using kubectl

Step 2. Download the tool again from: <https://ucs-migration-intl.obs.ap-southeast-3.myhuaweicloud.com/toolkits/k8clone-22.3.0.0329155427.tar.gz>

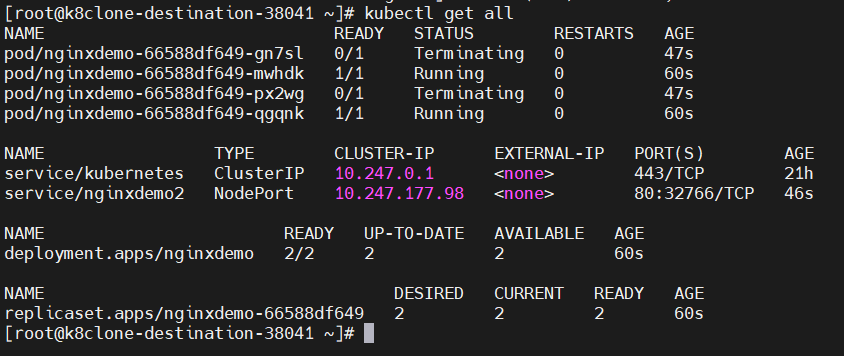
Step 3. Extract using:

$ tar -xvf [**k8clone-22.3.0.0329155427.tar.gz]**

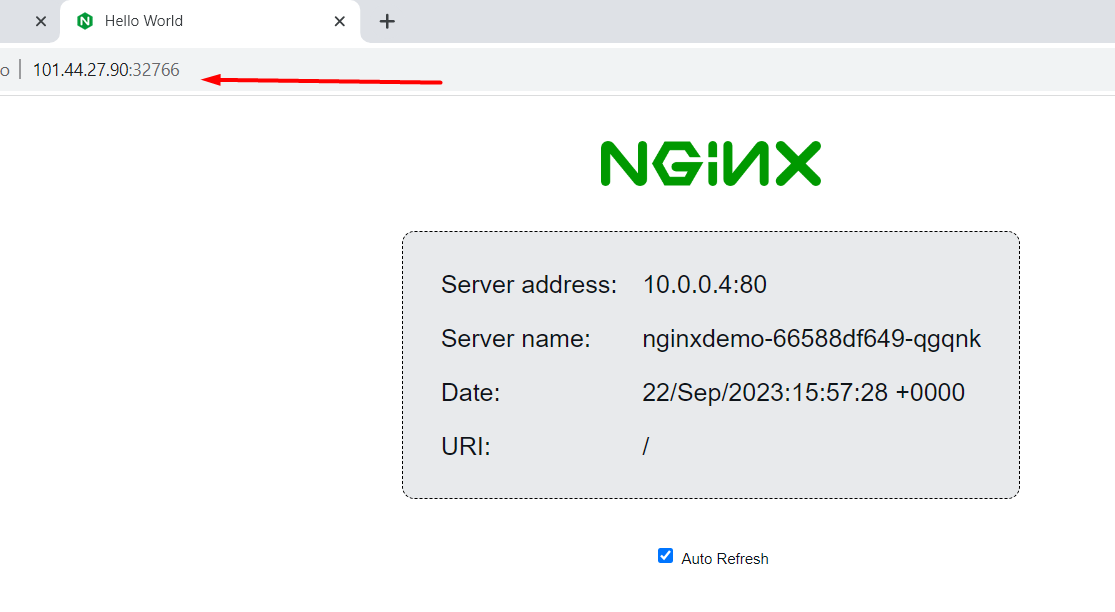
Step 4. Run the tool using the following command:

**$./**k8clone-linux-amd64 **restore -d ./k8clone-dump.zip -f ./restore.json**

Step 5. Run kubectl get all to confirm if all elements where restored.



Step 6. Using a local browser connect to node, or using an External IP to connect to the nginx demo to see that is running correctly and the application has correctly been migrated.



The node where the demo is running has a different IP that the one from the source cluster.

# FAQ:

* **When using the restore command, after it finishes several errors are also listed but the application still works.**

**-** *Timestamping or logs from the original application may be invalid to replicate on the destination cluster.*

**-***IP addresses tied to pods, will error as they may not be in the same range.*

* **When trying to execute the tool got Permission denied.**

**-** *Remember to change permissions of the tool using* ***chmod u+x****.*

* **My application is not immediately available after cloning.**

**-** *The deployment for your application needs to be exposed again in the context of the new cluster.*