IBM Cloud Kubernetes Cheat Sheet

IBM Cloud Kubernetes Service (IKS)

Get The IBM Cloud CLI Installer	
curl -sL https://ibm.biz/idt-installer | bash	
Login To IBM Cloud	**List All Worker Pools In A Cluster**
ibmcloud login -sso	ibmcloud ks worker-pools --cluster *cluster name or ID*
List Plugins	**List Available Regions**
ibmcloud plugin repo-plugins -r "IBM Cloud"	ibmcloud ks regions
Install A Plugin	**List All Availability Zones In A Region**
ibmcloud plugin install kubernetes-service	ibmcloud ks zones [--region-only]
Kubernetes (k8s) Service Help	**List Available Machine Types For A Zone**
ibmcloud ks --help	ibmcloud ks machine-types --zone *zone name*
Target A Region For K8s	**List Available Public And Private VLANs For A Zone**
ibmcloud ks region-set	ibmcloud ks vlans --zone *zone name*
View The Currently Targeted Region	**Create A Cluster**
ibmcloud ks region	ibmcloud ks cluster-create \
List All Clusters In A Region	--name *cluster name* \
ibmcloud ks clusters	--zone *zone* \
View The Details Of A Cluster	--machine-type *machine type* \
ibmcloud ks cluster-get *cluster name or ID*	--hardware <shared or dedicated> \
List All Worker Nodes In A Cluster	--workers *number* \
ibmcloud ks workers --cluster *cluster name or ID*	--public-vlan *public_VLAN_ID* \
View The Details Of A Worker Node	--private-vlan *private_VLAN_ID*
ibmcloud ks worker-get --worker *worker ID*	**Get The Command To Set The Environment Variable**
	ibmcloud ks cluster-config --cluster <cluster_name or ID>

Docker CLI

Log In To A Docker Registry	**List The Running Containers**
docker login *registry:port*	docker ps [-a]
Pull An Image Or A Repository From A Registry	**Delete A Container**
docker pull *name:tag*	docker rm *container name*
Create A Tag TARGET That Refers To SOURCE_IMAGE	**Delete All Running And Stopped Containers**
docker tag *source_image[:tag]* *target_image[:tag]*	docker rm -f $(docker ps -aq)
Push An Image Or A Repository To A Registry	**View The Details Of A Worker Node**
docker push *repo/image-name:tag*	docker exec -it *container name* /bin/bash
Run A Container	**Print The Last 100 Lines Of Logs**
	docker logs --tail 100 *container name*
docker run \	**Follow Log Output For A Given Container**
[--rm] \ Remove Container After It Exits	docker logs -f *container name*
[-it] \ Connect The Container To Terminal	**List The Docker Networks**
--name *assign a name* \	docker network ls
-p *private:public-port* \ Expose A Port	**List Docker Images**
[-v *path*] \ Create A Persistent Mount Volume	docker images
image:tag \ Image From Which The Container Is Started	**Delete A Container Image**
[command] The Command To Run Inside The Container	docker rmi *image_id*
E.g. /bin/sh	**Delete All Container Images**
Stop One Or More Running Containers	docker rmi $(docker images -q)
docker stop *container name*	
Kill One Or More Running Containers	
docker kill *container name*	

Helm

Install Helm Server-side Component	
helm init	
List Releases	
helm list	
List Installed Charts	
helm ls	
Search For An Installed Chart	
helm search *chart*	
Inspect & Prints The Contents Of A Chart	
helm inspect *chart*	
Install A Chart	
helm install --name *name* *chart*	
Delete A Chart	
helm delete --purge *chart*	

Kubernetes CLI

List All Namespaces	kubectl get namespaces
Set The Namespace Context	kubectl config set-context --current --namespace=<namespace>
List All Services In The Namespace	kubectl get services
List A Particular Deployment	kubectl get deployment *deployment*
List All Pods In All Namespaces	kubectl get pods --all-namespaces
Get All Running Pods In The Namespace	kubectl get pods --field-selector=status.phase=Running
Get A Pod's YAML	kubectl get pod *pod* -o yaml
List All Pods With More Details	kubectl get pod *pod* -o yaml --export
Describe A Pod	kubectl describe pods *pod*
List Pods Sorted By Restart Count	kubectl get pods \ --sort-by='.status.containerStatuses[0].restartCount'
Et The Version Label Of All Pods With Label App=*Label*	kubectl get pods --selector=app=*label* rc -o \ jsonpath='{.items[*].metadata.labels.version}'
List Worker Nodes	kubectl get nodes
Describe A Worker Node	kubectl describe nodes *node name*
Get The External IP Of All Nodes	kubectl get nodes -o \ jsonpath='{.items[*].status.addresses[?(@.type=="ExternalIP")].address}'
List Events Sorted By Timestamp	kubectl get events --sort-by=.metadata.creationTimestamp
List Containers In A Given Pod	kubectl describe pod *pod* --namespace *namespace* \ |grep container -B1
Follow The Logs From A Given Pod	kubectl logs -f *pod*
View The Logs From A Pod Since A Time	kubectl logs --since=2h *pod*
View The Logs Of A Specific Container In A Pod	kubectl logs --since=2h *pod* -c *container* \ --namespace *namespace*
Apply A Configuration To A Resource	kubectl apply -f *filename*

IBM Cloud Internet Services (CIS)

Install The CIS Plugin	
ibmcloud plugin install cis	
List All CIS Service Instances	
ibmcloud cis instances	
List Domains For A Given Service Instance	
ibmcloud cis domains --instance *instance name*	
List All Load Balancers For The Given Domain	
ibmcloud cis glbs *domain id* --instance *instance*	
List All GLB Pools For A Given Service Instance	
ibmcloud cis glb-pools --instance *instance*	
List Status Changes From Origins Connected To A Glb	
ibmcloud cis glb-events --instance *instance*	
Show The Details Of A Given GLB Pool	
ibmcloud cis glb-pool *pool id* --instance *instance*	
Show The Details Of A Given GLB Pool As JSON	
ibmcloud cis glb-pool *pool id* --instance *instance* -- output JSON	
Update A GLB Under A Given DNS Domain	
ibmcloud cis glb-update <domain id> <pool id> --json- file <json file> --instance <instance>	

IBM Cloud Container Registry

Install The CIS Plugin	
ibmcloud plugin install container-registry	
Set A Target Region	
ibmcloud cr region-set	
Log The Local Docker Client Into Registry	
ibmcloud cr login	
Displays All Images	
ibmcloud cr image-list	
Inspect One Or More Images	
ibmcloud cr image-inspect *image*	
Create A New Tag	
ibmcloud cr image-tag *source image* *target image*	
Delete One Or More Images	
ibmcloud cr image-rm *image*	
Build A Docker Image	
ibmcloud cr build [--no-cache] [--pull] [--quiet | -q] [--build-arg *key=value*] [--file *file*] | - f *file*] --tag *tag*	

Calico

List All Policies	
calicoctl get policy	
List A Policy In YAML Format	
calicoctl get -o yaml policy *policy.yaml*	
Create A Policy From A File	
calicoctl create -f *policy.yaml*	
Apply A Policy From A File	
calicoctl apply -f *policy.yaml*	
Replace A Policy From A File	
calicoctl replace -f *policy.yaml*	
Delete A Policy From A File	
calicoctl delete -f *policy.yaml*	
Delete Policy By Name	
calicoctl delete policy *name*	
Check The Status Of A Node	
calicoctl node status	