Module 2 - Monitoring Systems and Techniques

Module 2 Cheatsheet: Monitoring Systems and Techniques

Package/Method	Description	Code Example

bash	Bash, or the Bourne Again Shell command, is a command-line interpreter commonly used in Unix-based operating systems. It runs in a text window where the user can interpret commands to carry out various actions.	`#!/bin/bash`
		`#This is a Bash script that prints "Hello, World!"`
		`echo "Hello, World!"`

docker build	Builds Docker images from a Dockerfile and a "context" A build's context is the set of files located in the specified path or URL. .	`docker build -t your-image-name .` Replace your-image-name with the desired name for your Docker image. The . at the end of the command indicates that the build context is the current directory (where the Dockerfile is located).

docker network	You can use this code to manage networks. The subcommands can be used to create, inspect, list, remove, prune, connect, and disconnect networks.	`# create a docker network`
		` docker network create my_network`
		`#Verify Network Connection`
		`docker network inspect my_network`
		`#List docker Network`
		`docker network ls`
		`#Remove docker network`
		`docker network rm NETWORK_NAME_OR_ID`
		`#Prune docker network`
		`docker network prune`
		`#Connect Docker Network`
		`docker network connect NETWORK_NAME CONTAINER_NAME_OR_ID`
		`#Disconnect Docker Network`
		`docker network disconnect NETWORK_NAME CONTAINER_NAME_OR_ID`

docker ps	Lists the running containers by default. We can use different flags to get the list of other containers that are in stopped or exited status.	`docker ps [OPTIONS]`
		If you want to see all containers, including the stopped ones, you can use the -a or -all
		`docker ps -a`

docker pull	You can download Docker images from the internet.	`docker pull [OPTIONS] IMAGE_NAME[:TAG]`
docker run	It runs a command in a new container, getting the image and starting the container if needed.	`docker run [OPTIONS] IMAGE [COMMAND] [ARG...]`
docker stop	You can specify and stop one or more running Docker containers.	`docker stop CONTAINER_NAME_OR_ID`
grep	It searches for the pattern specified by the Pattern parameter and writes each matching line to standard output. The patterns are limited to regular expressions in the ed or egrep command style. The grep command uses a compact non-deterministic algorithm.	`grep [OPTIONS] PATTERN [FILE...]`

		Example: Search for a pattern in a file. `grep "search_pattern" file.txt`
git clone	You can create a copy of a specific repository or branch within a repository.	`git clone REPOSITORY_URL [DESTINATION_DIRECTORY]`

oc	It is the main command line for OpenShift. It comprises tools to build, deploy, and administer containers.	`oc login [OPTIONS]`
		`oc new-project PROJECT_NAME`

oc apply	It computes and applies differences between objects by performing a three-way merge between the input into the command, the current version of the object, and the most recent user-specified object definition stored as an annotation in the current object.	`oc apply [OPTIONS] -f FILE_PATH`
oc create deployment	Helps create a deployment with the specified name.	`oc create deployment DEPLOYMENT_NAME --image=IMAGE_NAME`

oc describe	This operation can be used to return detailed information about a specific object□example, pods.	`oc describe RESOURCE_TYPE RESOURCE_NAME`
		`Example: oc describe pod my-pod`

oc expose	Exposes a replicated application as a service or route.	`oc expose SERVICE/POD [OPTIONS]`
		`Example: oc expose pod my-pod`

oc get	Displays one or many resources for example, pods, replication controllers, etc.	`oc get RESOURCE_TYPE [OPTIONS]`
		`Example: oc get pods`

touch	Updates the access and modification times of a file.	`touch [OPTIONS] FILENAME`
		`</td>`

Skills Network