**Справочник команд к курсу DevOps**

* **Git**
  + git init – инициализация репозитория в текущей директории
  + git status – текущее состояние файлов локального репозитория
  + git add . / git add -A – добавление файлов в трекинг git репозитория
  + git commit -am “commit name” – добавить изменённые файлы в трекинге в stash и закоммитить изменения
  + git branch new\_branch – создать ветку new\_branch
  + git checkout new\_branch – переключиться на ветку new\_branch
  + git checkout -b new\_branch – создать ветку new\_branch и переключиться на неё
  + git branch – посмотреть ветки в локальном репозитории, текущая будет помечена \*
  + git merge new\_branch – влить ветку new\_branch в текущую ветку
  + git log – посмотреть список коммитов
  + git diff file1 – посмотреть изменения в file1 относительно остальных коммитов
  + git clone <http://gitlab.loc/repo_name> - склонировать репозиторий с сервера git
  + git push - запушить локальные коммиты на сервер git
* **Docker**
  + docker build -t my\_image:1.0 . – собрать образ my\_image:1.0 из Dockerfile в текущей директории
  + docker ps – посмотреть запущенные контейнеры
  + docker ps -a – посмотреть все созданные контейнеры
  + docker run --name=web -p 80:80 -v /tmp/local:/tmp/container -d nginx:latest – запустить контейнер с именем web в detach режиме (-d) из образа nginx:latest, пробросить порт 80 наружу контейнера, пробросить локальную директорию /tmp/local в контенер /tmp/container
  + docker exec -it -u root web sh – запустить sh консоль от рута внутри контейнера web и переключиться на неё
  + docker inspect web – посмотреть конфиг контейнера web в json формате
  + docker logs web – посмотреть stdout контейнера web
  + docker logs -f --tail=100 web – посмотреть stdout контейнера web с параметрами, аналогично

tail -f -n 100

* + docker stop web – остановить контейнер web
  + docker rm web – удалить контейнер web
  + docker rm -f web – принудительно удалить контейнер web без остановки, аналогично kill -9
  + docker images – посмотреть список локальных образов
  + docker rmi nginx:latest – удалить образ nginx:latest
  + docker network ls – посмотреть список сетей docker
  + docker network create mynet – создать сеть mynet
  + docker network rm mynet – удалить сеть mynet
* **Docker-compose**
  + docker-compose up -d --build – собрать образы и запустить контейнеры в detach режиме, описанные в файле docker-compose.yaml
* **Ansible**
  + ansible-playbook playbooks/docker.yaml -u admin -kK -b – запустить плейбук playbooks/docker.yaml, для ssh соединения использовать юзера admin, спросить ssh и sudo пароли
* **Kubernetes**
  + kubectl get nodes – посмотреть список нод в кластере kubenetes
  + kubectl get nodes -o wide – посмотреть список нод в кластере kubenetes в расширенном варианте
  + kubectl get ns – посмотреть список namespaces в кластере kubenetes
  + kubectl create ns myns – создать namespace myns
  + kubectl -n myns get pods – посмотреть список подов в namespace myns
  + kubectl get pods – посмотреть список подов в namespace default
  + kubectl get pods -A – посмотреть список подов во всех namespace
  + kubectl get deployment/svc/ing/daemonset/statefulset – посмотреть список deployments/services/ingresses/daemonsets/statefulsets в namespace default
  + kubectl delete po/deployment/svc/ing/daemonset/statefulset myname – удалить pod/deployment/service/ingress/daemonset/statefulset myname в namespace default
  + kubectl get storageclass - посмотреть список storageclass в кластере
  + kubectl get pv - посмотреть список persistent volumes в кластере
  + kubectl get pvc -A - посмотреть список persistent volume claims во всех namespace
  + kubectl -n myns get ing nginx-ingress -o yaml – вывести yaml конфиг ingress в namespace myns
  + kubectl -n myns apply -f deployment.yaml – создать объекты из файла deployment.yaml в namespace myns, так создаются любые объекты, deployment, service, storageclass и т.д.
* **Helm**
  + helm -n myns list – посмотреть список helm deploys в namespace myns
  + helm -n myns upgrade -i myapp . – обновить helm deploy myapp в в namespace myns, если myapp отсутствует, то создать его
  + helm -n myns delete myapp – удалить helm deploy myapp в в namespace myns