Why choose the DevOps methodology

Advantages

- Faster release cycle as a result of the continuous integration/continuous delivery.
- Transition from capital expenses to operational expenses where the company no more invests in buying hardware but uses a cloud provider where the expenses are pay per use (Capex Vs Opex)
- Applications can have zero downtime when a new release takes place through the use of container orchestration tools like Kubernetes and rolling deployments.
- Applications are easy to scale because of the flexibility either through k8s or the cloud provider.
- No more issues because of the difference between libraries on an OS and the other as docker allows the application to be fully packaged with their dependencies.
- Easier management for servers and cloud infrastructure through the use of configuration management and infrastructure as code.

Tools used

HashiCorp
Packer

docker

OPENSHIFT
by Red Hat

ANSIBLE

Configuration Management

- RedHat ansible is considered the De-Facto for configuration management, it’s also capable of being used with cloud providers to provide IaC capabilities.
- The main advantage of ansible is that it doesn’t need to be installed on any of the managed hosts, all there needs to be is ansible running on the controller node and python being installed on all the hosts, ansible relies on SSH connection to run the playbooks on the managed nodes.
- If we relied on the use of ansible modules aside from “shell, command and raw” modules, then the playbooks written should be idempotent (They give the exact same result regardless of how many times we run them)
- Other configuration management tools in use are chef, puppet and SaltStack.

Infrastructure as Code

- Declarative way for specifying the resources needed to be used from the cloud provider
- The most famous tool is Terraform from HashiCorp which uses the company’s own language for creating the files (HCL)
- Ansible can be used to do the same job with the provided cloud modules
- New alternatives exist like Pulumi which uses programming languages to do the job

Containerization

- The process of placing the application with its dependencies inside a container that is placed directly on the host OS
- Relies on a container runtime which manages the containers, the most famous one is docker but alternatives are already out there like Podman, rkt and containerd.
- Containers are lightweight unlike VMs, they allow for tight control over the resources through the use of cgroups, processes running inside the container are also separate from the host OS through the use of namespaces.

Container image builders

- While docker provided capabilities to build the image and run it, currently there are tools specific to build the container images and they are OCI compliant.
- Examples include Kaniko by Google, Makisu by Ubder, Buildah which is being used alongside Podman and skopeo in RHEL 8
- Kaniko and Makisu’s main advantage is that they build images from Dockerfiles without the need for root access thus they’re more secure.

Automated Machine Images

- The previous builders mentioned are used in container world, Packer deals with the servers and allows the creation of already pre-configured images.
- Instead of running an ansible playbook on an ec2 instance packer can do the job using ami builder in combination with ansible provisioner to create an already pre-configured ec2 image that can be directly used on AWS
- This saves time in case the configuration for the instances take an already long time

HashiCorp
Packer

Container Orchestration

- While docker is able to create containers there’s no guarantee that the containers will always run, errors might happen and there needs to be a way of managing the containers.
- Kubernetes is the De-facto for container orchestration, it’s a complex tools with rich features that allows container management, containers are placed inside pods, usually one container per pod but there are use cases for multi container pods and there are known patterns like sidecar, ambassador and adaptor.
- Kubernetes ensures that pods keep running through the use of replication controllers that can spin up new pods so that the state is always matching the desired replicas.
- RedHat built OpenShift on top of Kubernetes, RedHat also integrated most CoreOS features into OpenShift.
- Alternatives for K8s exist like Docker swarm, Apache Mesos and HashiCorp’s Nomad but K8s remains dominant

HELM

OPENSHIFT
by Red Hat

HashiCorp
Nomad

CI/CD

- Continuous integration originated from the days of Extreme Programming and it’s made as a result of the lack of communication between software developers which caused the integration phase to take very long time on its own.
- The purpose of CI is to fail quickly, if there’s anything wrong with the newly added code then the build should fail as soon as possible and a fix should be made, each push should trigger a new build.
- Continuous delivery and continuous deployment are close to each other, however with continuous deployment if the build is successful then it gets deployed without no intervention so that’s more like 100% automation.
- Jenkins from CloudBees is a very famous tool for continuous integration while Spinnaker from Netflix is used for continuous deployment.