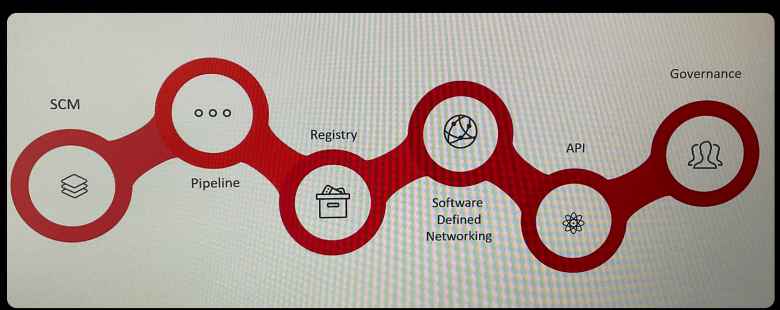
• Openshift 4

- Openshift Origin
- Openshift Online
- Openshift Dedicated
- Openshift Enterprise

- Openshift origin —

  - Tools in Openshift



- Openshift works on top of kubernetes

Openshift config vs kubernetes config

k8s	vs	Openshift
① ReplicaSet		Replica Controller
set-based		Equity-based

↓ k8's
— use **Deployment** instead of **DeploymentConfig** (openshift config)

- Image stream

Way to automatically fetch images from set location when there is newer update to it.

Build —
creating an image from the source

BuildConfig - is how the configuration of that Build is
→ Done automatically by using Build strategies.

Build strategy

1. Source to Image
2. Pipeline → for automation via Jenkins
3. Docker
4. Custom

Build Inputs

1. Git
2. Binary
3. Dockerfile
4. Image
5. Input Secrets
6. External Artifacts

- To build OCI compliant dockerfiles we use **buildah** tool

S2I - Source to Image

- Secure
- Faster build time
- Easily patch the build.

- Build triggers in Openshift

① Webhook – Send request to API

② Image Change -
tip :- Don't set image change for dev env (increases management if large no. of daily changes)

③ Configuration change -

- Networking

- Openshift SDN handles the openshift network

Default Network IP Address
– 10.128.0.0/14

- oc get pods -o wide

- TO see the network IP's.
  - pods restart may change the IP add.
  - that's its always better to use service names

- Services in Openshift

Same as kubernetes

- Works as Load Balancer

- Service has internal IP's assigned for internal communication & service are linked to pods using selectors.

- Routes

- Are used to connect Services outside the network

With using Routes we can
-① Load Balance
② Security
③ Split traffic

Load

Routing strategy

① source strategy – same server every time

② round-robin – different server every time

③ leastconn – connect to lowest conn

Source strategy for Load Balancing

– In Techincal term its just a sticky session

- AutoScaling

① Through cmd

② Through YAML file

① `oc autoscale deployment/front-end --min=2 --max=7`

- storage

- Persistance storage

for this there are multiple different plugins in Openshift

ex. AWS EBS.

- oc get sc

To see our storage class

- CPU / RAM Allocation

- Ask CPU, Memory
- Limit → Limit crossed App goes down

for CPU → Set Not limit
for Memory → Set limit

Resource Quota
— To add limits to the amount of resources that Application have
example yaml

—

namespace.yaml ● ! apiVersion: v1 Untitled-1 ●

io.k8s.api.core.v1.ResourceQuota (v1@resourcequota.json)

```yaml
apiVersion: v1
kind: ResourceQuota
metadata:
  name: core-object-counts
spec:
  hard:
    configmaps: "10"
    persistentvolumeclaims: "4"
    replicationcontrollers: "20"
    secrets: "10"
    services: "10"
    services.loadbalancers: "2"
```

oc get resourcequota -n limittester

- Example of MicroServices on Openshift

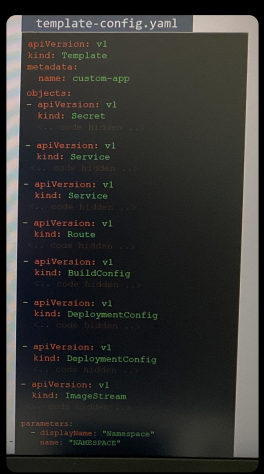
- Openshift templates and catalog

To create your own templates

- kind: Template

Create different sections for Deployments, Service, route, Imagestream Namespaces

ex.



- oc create -f yamlfile.yaml.

You can create your custom templates which can have whole application deployed with single yaml file.

- User management
  - oc create user sam
  - oc create identity allow_all: sam
  - oc create useridentitymapping allow_all: mike

  oc delete user sam

— Service Accounts - Used to manage, create and delete Accounts

- oc create sa samService
- oc get sa
- oc policy add-role-to-user view system: serviceaccount: sockshop: samService

- Groups

- oc adm groups new samgroup sam bob

  Two users

° RBAC – Roles based Access control

What is RBAC -

① Authorization to resources
② Out-of Box Security

Roles attached to groupes
- We can restrict groups to view pods in particular namespace only.

• cluster role
  Scoped to entire cluster

In Yaml file of Roles in Openshift

• To have role for cluster remove namespace from yaml
  kind: ClusterRole

Example of creation of Role

① create a role

② Create a Service
③ Create a RoleBinding for both

- ConfigMap - Same as k8's

We can pass any type of config data to the containers at runtime using ConfigMaps

- oc create configMap name
- oc get configMap

- Secreat password -
  → Same like AWS Secreat password.

Secreats - Encrypted Passwords and all

SCC - Security Context Constraints

Network policy -
↑
Inbound / outbound traffic

Routers
Switches
firewalls
Gateways

- SSC - is about privileged access control its about permissions for the pod itself

Open policy agent & SSC is kind of doing similar thing
- but SSC is build in