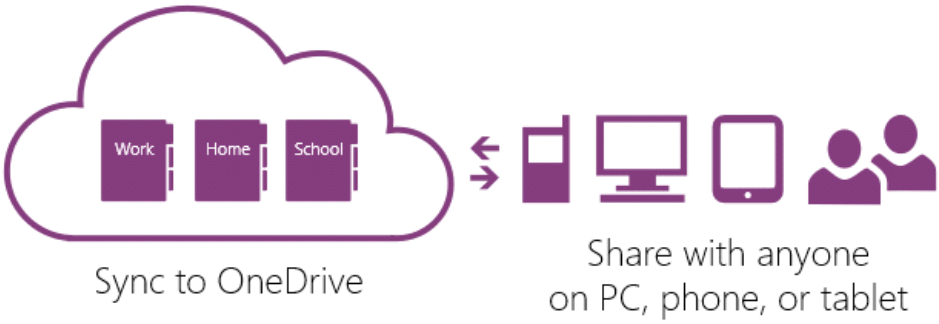
OneNote: one place for all of your notes



Watch the
2 minute video

1. Take notes anywhere on the page

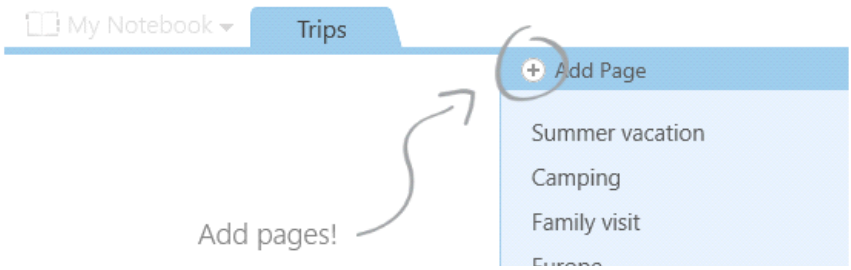
Write your name here

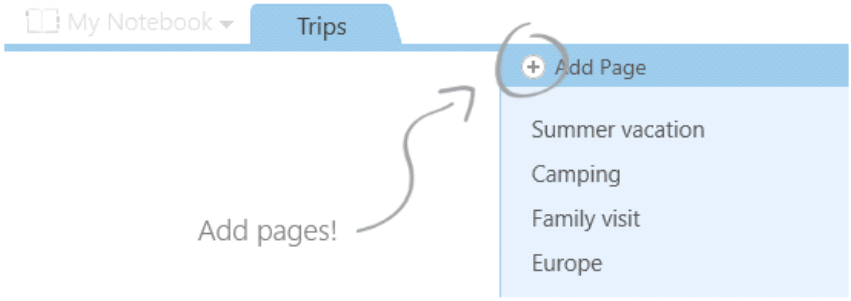
2. Get organized

You start with "My Notebook" - everything lives in here

Add sections for activities like:

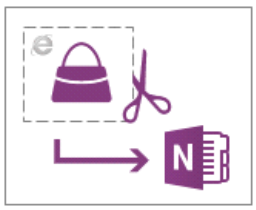
Add pages inside of each section:



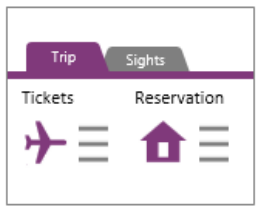


(Pages are over there)

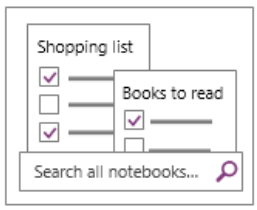
3. For more tips, check out 30 second videos



Clip from the web



Plan a trip with others

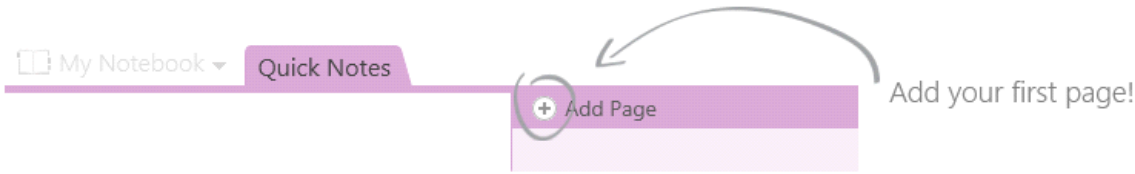


Search notes instantly

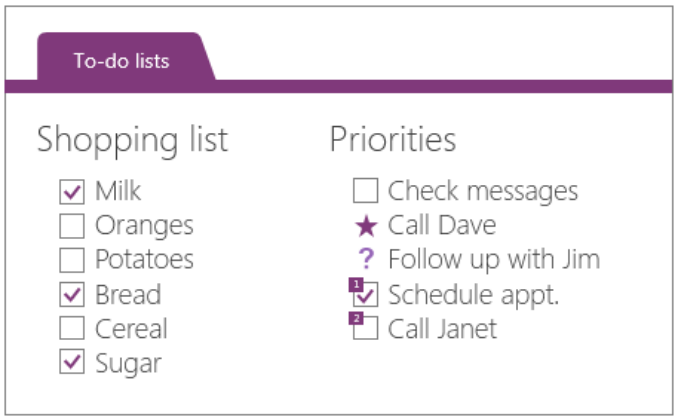
Write notes on slides

4. Create your first page

You're in the Quick Notes section - use it for random notes

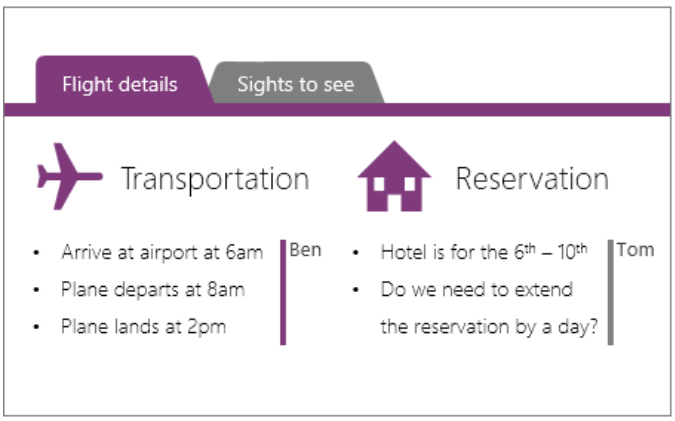


OneNote Basics



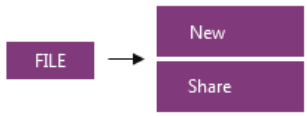
Remember everything

- Add Tags to any notes
- Make checklists and to-do lists
- Create your own custom tags



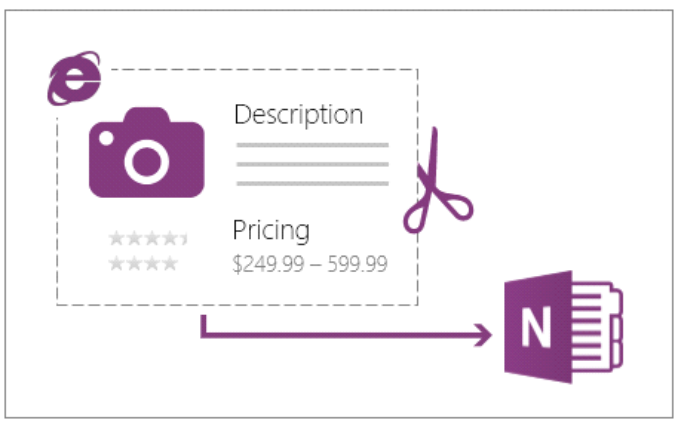
Collaborate with others

- Keep your notebooks on OneDrive
- Share with friends and family
- Anyone can edit in a browser



Keep everything in sync

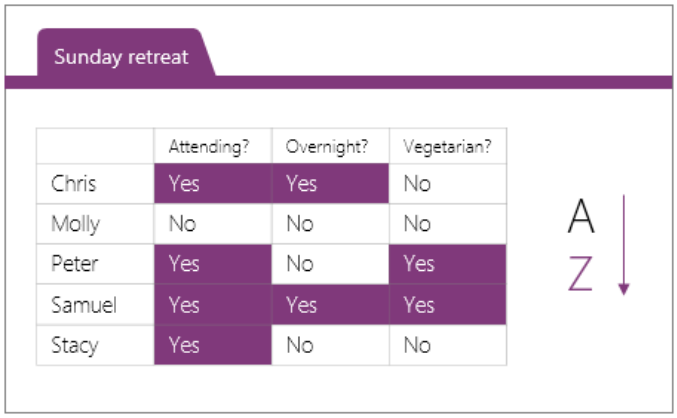
- People can edit pages at the same time
- Real-Time Sync on the same page
- Everything stored in the cloud
- Accessible from any device



Clip from the web

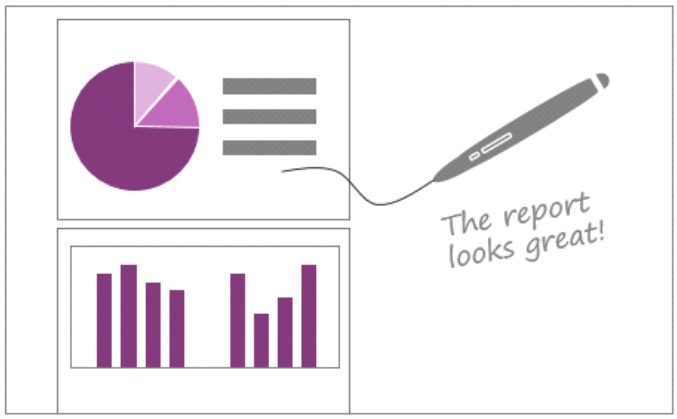
- Quickly clip anything on your screen
- Take screenshots of products online
- Save important news articles

in your taskbar
OR
+ S on your keyboard



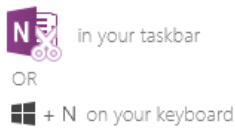
Organize with tables

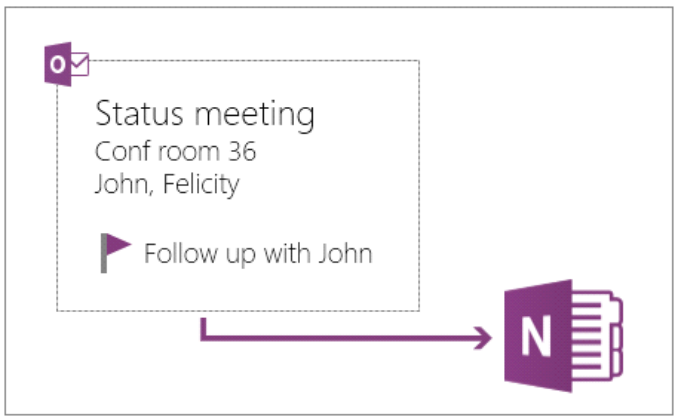
- Type, then press TAB to create a table
- Quickly sort and shade tables
- Convert tables to Excel spreadsheets



Write notes on slides

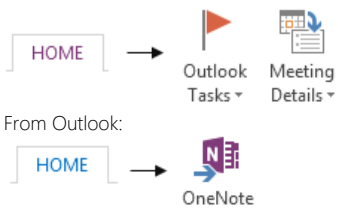
- Send PowerPoint or Word docs to OneNote
- Annotate with a stylus on your tablet
- Highlight and finger-paint

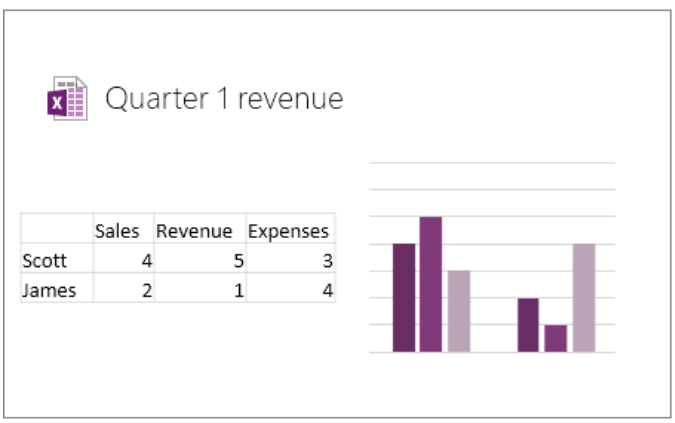




Integrate with Outlook

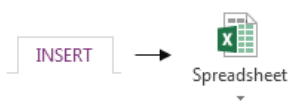
- Take notes on Outlook or Lync meetings
- Insert meeting details
- Add Outlook tasks from OneNote

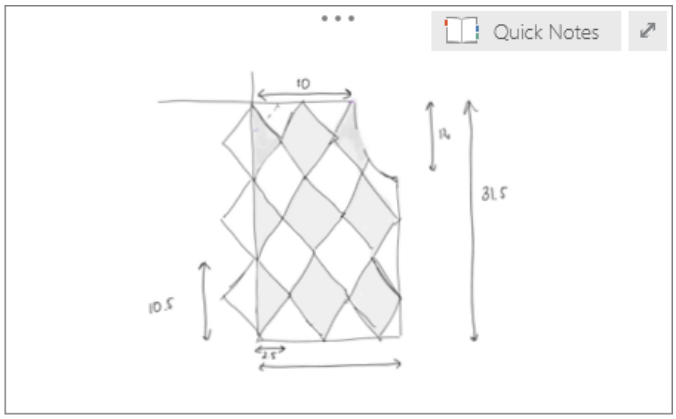




Add Excel spreadsheets

- Track finances, budgets, & more
- Preview updates on the page

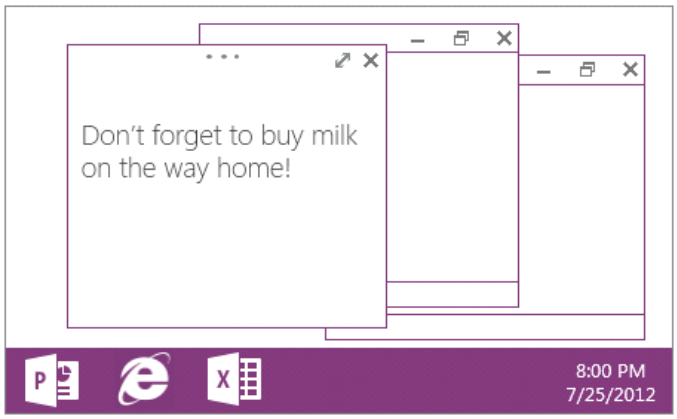




Brainstorm without clutter

- Hide everything but the essentials
- Extra space to focus on your notes

in the top corner of the page



Take quick notes

- Quickly jot down thoughts and ideas
- They go into your Quick Notes section

in your taskbar

OR

+ N on your keyboard

1. Pushing the files to the repository.(NEIOTAPPFM-106) -- done
2. Create one more directory and push the files related to custom postgres image --- done
3. Creating the base image using nokia registry base images i.e., postgres base image -- done
4. Pushing the base images to the nokia internal registry instead of docker hub -- done
5. Upgrading the versions of washoutdataload dynamically using latest as tag

21 February 2019 11:55

git pull
Git status
git add file1 file1 fil3 ....
git commit -m "NEIOTAPPFM-26:Dockerized deployment"
git push origin HEAD:refs/for/master (for review) or git push

Fm docker build

21 March 2019 13:10

Delete all stopped container - docker rm $(docker ps -a -q)

docker build --build-arg http_proxy="http://10.158.100.6:8080" --build-arg https_proxy="http://10.158.100.6:8080" -t si-server-base:19.0 -f DockerFileBaseImage .

docker build --build-arg http_proxy="http://10.158.100.6:8080" --build-arg https_proxy="http://10.158.100.6:8080" -t si-server-apps:19.4 -f DockerFileWebApps .

docker tag si-server-apps:19.4 sandbox-docker-inprogress.repo.lab.pl.alcatel-lucent.com/si-server-apps:19.4

docker push sandbox-docker-inprogress.repo.lab.pl.alcatel-lucent.com/si-server-apps:19.4

Jupyter link:http://localhost:8888/notebooks/Documents/DataScience/vulnerability_model/vulnerabillity_input_model.ipynb

28 February 2019 20:33

https://confluence.app.alcatel-lucent.com/x/pyDxD

https://nokia-my.sharepoint.com/:w:/r/personal/jaisuthan_amaldass_nokia_com/_layouts/15/WopiFrame.aspx?sourcedoc=%7B71797B0D-127B-4136-B752-3F2EEBF1815C%7D&file=WaterApplication.rtf&action=default

VulnerabilityInputCreate

https://confluence.app.alcatel-lucent.com/display/AAIOT/Data+Science+Module+Interface

```
java -cp postgresql-42.2.5.jar:h2o.jar water.H2OApp
```

From <https://h2o-release.s3.amazonaws.com/h2o/rel-vajda/1/docs-website/h2o-docs/getting-data-into-h2o.html>

docker run -it --rm --name graal --link rabbitmq-mgmt -p 10080:8080 -v C:\Users\vvignesw\Documents\DataScience\vulnerability_model\spring-r\target:/home/test herrybud/graal:1.0 java -jar /home/test/spring-r-0.0.1-SNAPSHOT.jar

mvn -Dmaven.test.skip=**true** clean install

select count(1) from operation.historic_events;
select count(1) from water_common.rail_elevations;
select count(1) from water_common.rail_grid_cells_info;
select count(1) from water_common.alluvial_fans;
select count(1) from water_common.culverts;

https://fm-dev1.iot.nokia.com/washoutdataload/jobs/HistoricEventsLoad
https://fm-dev1.iot.nokia.com/washoutdataload/jobs/RailGridCellsInfoLoad
https://fm-dev1.iot.nokia.com/washoutdataload/jobs/VulnerabilityInputCreate

06 March 2019 15:26

Istio - Service Mesh, openshift
https://www.openshift.com/learn/topics/service-mesh
https://www.digitalocean.com/community/tutorials/how-to-install-and-use-istio-with-kubernetes

Acka Model - Actor Model
Serverless Architecture
Domain Driven
Event Sourcing, Event
Stateless
gRPC over HTTP2
graphQL
http://camunda.com/
Chaos engineering
Protocols - MQTT, LoRA

Spark
RDD and Data Frames.

- ✓ Spark SQL.
- ✓ Spark Optimization Techniques.
- ✓ Understanding of Yarn/Spark architecture.
- ✓ Hive & HDFS.
- ✓ Python Programming.
- ✓ UNIX/Shell Scripting.

AWS certified developer - Associate - https://app.pluralsight.com/paths/certificate/aws-certified-developer-associate

Distributed Caching - Redis Sentinel

Hazelcast - https://app.pluralsight.com/library/courses/hazelcast-getting-started/table-of-contents
Java process builder
rtmp protocol
Gstreamer
AWS (EC2, RDB, Monitoring)

Kubernetes

04 May 2019 19:52

- Features
  - Automatic Rolling upgrade and rollback
  - Automatic scheduling of pods to nodes based on command
  - Horizontal Scaling
  - Load Balancing
  - Self healing capabilities - restarting the containers which are died, rescheduling, replacing
  - Master/Worker nodes

- Masters
  - Scheduler
    - Responsible for scheduling/allocating the pods to the requirements/affinity of the nodes
    - Tracking the utilization of work load on cluster nodes and placing the work load on the resources are available and accepts the workload
  - Controller Manager
    - Check whether the current state matches with desired state
    - Eg: Replication controller, namespace controller, endpoint controller, ingress controller(nginx)
    - Collectors regulates the state of cluster and performs a task
  - Etcd -- key value store
    - Highly available key value store
    - It is accessed only by API server since it has sensitive information and also by kubelet to allocates the task based on the instructions
  - ApiServer -- accepts yml(internally converts yml to json) and Json
- Nodes
  - Kubelet
    - Basic service in each node responsible to transfer information to and from control plane service
  - Container Runtime
  - Kube-proxy
    - Proxy service which run on each node which helps in making service available to external host
    - Forwarding the request to respective nodes to do load balancing
- Pods -- atomic unit
- Services --
  - ClusterIp - cluster internal communication
  - NodePort - external exposing of ports
  - LoadBalancer
    - Label with selector
    - Label without selector -- for access of external resources likes database by explicitly creating endpoints with ipaddress and port of resources
- Deployments
  - BlueGreenDeployments
  - Rolling Deployment
- Secrets
- ConfigMap
- Replication Controller -- replicasets in kubernetes(ensure that the desired state matches the current state)

- Ports, TargetPorts, NodePort
  - Port - port on which services can be accessed from other services in a cluster
  - Nodeport - port on which services can be accessed externally using kube-proxy
  - TargetPort - port on which pod will be running but the service port will be different
  - https://stackoverflow.com/questions/49981601/difference-between-targetport-and-port-in-kubernetes-service-definition
  - https://matthewpalmer.net/kubernetes-app-developer/articles/kubernetes-ports-targetport-nodeport-service.html

- Ingress Controller -- runs of on top of Services for reverse proxy and DNS name and path configuration.
  - https://dzone.com/articles/ingress-controllers-for-kubernetes

Commands:

S.no	commands	explaination
1	kubectl get nodes	to know the status of nodes
2	kubectl get deployments	to know the status of deployments
3	kubectl get events	to know the status of nodes
4	kubectl get pods	
5	kubectl get all	
6	kubectl create -f helloworld.ym	To create the new pod, services
7	kubectl rollout status deployment.v1.apps/helloworld	
8	kubectl rollout status deployment helloworld	
9	kubectl describe pods helloworld	
10	kubectl delete deployment/helloworld	
11	kubectl describe deployment helloworld	
12	kubectl delete pod/helloworld	
13	kubectl get event --field-selector involvedObject.name=<object name/id> Eg. helloworld-7bcf694f95-wwx22	
14	kubectl expose deployment helloworld --type=LoadBalancer --name=my-service	
15	kubectl describe services my-service	
16	kubectl delete service my-service	
17	kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v1.10.1/src/deploy/recommended/kubernetes-dashboard.yaml	To update the existing pods, services
18	kubectl proxy --port=8002	
19	kubectl get svc --namespace=kube-system	
20	kubectl get deployment --namespace=kube-system	
21	Kubectl config get-contexts	
22	Kubectl config current-context	
23	Kubectl config view	To view the cluster , context configuration
24	Kubectl config use-context cluster-name	
25	Kubectl get namespaces	To get the list of namespaces

Cheat sheet for commands - https://kubernetes.io/docs/reference/kubectl/cheatsheet/

https://kubernetes.io/docs/tasks/access-application-cluster/service-access-application-cluster/
https://kubernetes.io/docs/concepts/services-networking/service/
https://github.com/karthequian/docker-helloworld/blob/master/deployment.yml

Services
https://medium.com/google-cloud/kubernetes-nodeport-vs-loadbalancer-vs-ingress-when-should-i-use-what-922f010849e0
https://www.edureka.co/community/19351/clusterip-nodeport-loadbalancer-different-from-each-other

Creating persistent volumes - https://kubernetes.io/docs/tasks/run-application/run-single-instance-stateful-application/

- Volumes --
  - Shared Volume
    - cinder(openstack volume),
    - awsElasticBlockStore(ews volume),
  - Individual Volume
    - hostPath
    - ConfigMap to pass the config data values to the application using configuration file

InterPod Communication

- Database containers will have ClusterIP as Services and it will be referred with their service name from other pods
- Inter Application communication can happen thourgh NodePort, ClusterIp with the service names.

Data Plane vs Control Plane - https://blog.envoyproxy.io/service-mesh-data-plane-vs-control-plane-2774e720f7fc

Environmental Data can be added as env tag under the templates(pods) of the container configuration in pods

http://collabnix.com/kubernetes-dashboard-on-docker-desktop-for-windows-2-0-0-3-in-2-minutes/
https://www.digitalocean.com/community/tutorials/how-to-install-software-on-kubernetes-clusters-with-the-helm-package-manager

Sticky & Non-Sticky Sessions - https://stackoverflow.com/questions/10494431/sticky-and-non-sticky-sessions

Stateful sessions - https://kubernetes.io/docs/tutorials/stateful-application/basic-stateful-set/
Stateful set vs deployment - https://medium.com/stakater/k8s-deployments-vs-statefulsets-vs-daemonsets-60582f0c62d4
https://supergiant.io/blog/creating-stateful-apps-with-kubernetes-statefulsets/
Persistent volume in deployment and stateful set - https://akomljen.com/kubernetes-persistent-volumes-with-deployment-and-statefulset/

Deployment -
2 pods replicas sharing the same pvc,
only 1 pvc will be created and shared between pods
Creates replicaset first and then the pod
Rolling upgrades will be happening at controlled rate

StatefulSet
Each pod replicas will be having their own pvc created
Will not create any replicaset

Headless services -
no load-balancing is done and no cluster IP is allocated for this service
https://medium.com/faun/kubernetes-headless-service-vs-clusterip-and-traffic-distribution-904b058f0dfd

Nodeaffinity - hard/soft

PersistentVolumeClaims
readwriteOnce
readWriteMany
Resources:requests:50Mi

Use cases for multi-pod containers - https://www.mirantis.com/blog/multi-container-pods-and-container-communication-in-kubernetes/
Side car containers - log or data change watchers, monitoring adapters
Proxies, bridges & adapters

Yet to prepare:
Configuration - https://kubernetes.io/docs/concepts/configuration/overview/
https://www.mgasch.com/post/podnodesel/
https://www.digitalocean.com/community/tutorials/how-to-install-and-use-istio-with-kubernetes

kubectl delete pod --grace-period=0 --force

Helm Charts

Helm - package manager
Charts - Repository

Monitoring Tool

Helm Charts

08 May 2019 12:23

Helm charts will be used for packaging the container as charts and managing it through helm package manager.

Helm chart commands

S.no	commands	explaination
	kubectl -n kube-system get po	Get the pods running on kube-system namespace
	helm init --history-max 200	Creating the tiller(Server) in kubernetes cluster(current context)
	helm init --upgrade	Upgrading the tiller config
	helm reset --force	Deleting the tiller(server) from the kubernetes cluster
	helm ls	Listing out the chart deployments
	helm repo update	To pull the charts from central repo and keeping it in local repo
	helm install stable/mysql	Installing the charts and deploying it
	kubectl -n kube-system describe pod tiller-deploy-6ddf99784-8pmmr	Namespace(kube-system) describing the pods with name
	helm delete piquant-lobster	Delete the helm chart deployment
	Helm search local	To search charts in local repository
	Helm search	To search charts in central repository

Installing the Helm -- https://helm.sh/docs/using_helm/#installing-helm
Helm Charts general - https://kubernetes.io/blog/2016/10/helm-charts-making-it-simple-to-package-and-deploy-apps-on-kubernetes/

S.no	commands	explaination
1	helm install --dry-run --debug ./mychart	
2	helm create mychart	
3	Helm deploy	
	helm package ./mychart	Package it as tgz flie
	Helm serve	To run the local repository
	Helm ls	To view the releases
	Helm ls	
	Helm delete	To delete releases

BuiltIn Objects -
https://github.com/helm/helm/blob/master/docs/chart_template_guide/builtin_objects.md

Requirements.yml -- adding dependency of other components like database to main application through requirements yaml file.

helm install --name isa-citm citm-ingress-1.14.7.tgz -f deploy.yaml

fm-app.nokia.iot.com

CheatSheet

14 May 2019 15:10

Helm & Kubernetes Cheat Sheet

https://gist.github.com/tuannvm/4e1bcc993f683ee275ed36e67c30ac49#ifelse

Cassandra

28 May 2019 12:12

- Ref:https://docs.datastax.com/en/archived/cassandra/3.0/
- Different types of NoSQLs - https://www.3pillarglobal.com/insights/exploring-the-different-types-of-nosql-databases
    - Document based store - couchDB, mongoDB - stores in the form of JSON
    - Key-Value based store - redis, Riak, Amazon Dynamo DB - stores in the form of key(String) and value as String, JSON, BLOB etc..
    - Column Family Store - Hbase, Cassandra - stores by grouping the cells based on the columns instead of rows
- Data read
    - Read repair in case of mismatch in read acknowledgement

- Data will be written to
    - commit log (for crash recovery safety purpose)
    - MemTable
    - If the MemTable data reaches configurable limit, then it will be flushed out to SSTable
- Data deletion process
    - Data will be logically deleted
    - Once the DELETE command executed, data will be marked for deletion using **tombstone**, a marker.
    - Tombstone has a built in expiration time.
    - During compaction, using tombstone data will be logically marked as deleted, since data written to SSTable is immutable
    - **TTL** can be set for each records
    - **Zombie** records
    - **Grace period** for recovering node to update the deleted records as tombstone record
    - Tombstone cleaning - https://academy.datastax.com/support-blog/cleaning-tombstones-datastax-dse-and-apache-cassandra
- Consistencey Levels -
    - Strong consistency - R+W>N
    - Eventual consistency - R+W=<N
- https://docs.datastax.com/en/cassandra/3.0/cassandra/dml/dmlConfigConsistency.html
- https://docs.datastax.com/en/cql/3.3/cql/cql_reference/cqlshConsistency.html
- https://docs.datastax.com/en/cassandra/3.0/cassandra/dml/dmlConfigConsistency.html
    - Read Consistency - read consistency becomes read intensive operation because it will check for same data from multiple nodes
    - Write Consistency - If it is write intensive operation
        - Response will be sent to the user unless the write operation happened in commit log and mem cache table
        - Response will be collected from multiple nodes
- QUOROM - (replication_factor / 2) +1
    - Helps in verifying Consistency level
    - Minimum (these much node counts) has to be read/write before the response send to the node which receives the request
- Masterless architecture
- Coordinator node
- Examples of Consistency level - https://docs.datastax.com/en/cassandra/3.0/cassandra/dml/dmlClientRequestsReadExp.html
- CAP theorem
    - Consistency - data consistency - RDBMS best by doing it in transaction format
    - Availability -
    - Partition - For horizontal Scaling
        - Partition Key - to decide the node based on the partition key,

        - Clustering column - sort the row based on the clustering column in the each particular partition

    - Cassandra - combination of Partition & Availablity
      CREATE TABLE crossfit_gyms_by_location (
        country_code text,
        state_province text,
        city text,
        gym_name text,
        PRIMARY KEY (country_code, state_province, city, gym_name)
      );
      Note that only the first column of the primary key above is considered the partition key; the rest of columns are clustering keys
      Link for ref: https://shermandigital.com/blog/designing-a-cassandra-data-model/

- Column Families - by choosing the node based on partition key(column level)
- Hiereachy
    - Machine/Node --> KeySpace --> Column Family(Table) --> Partition
    - Keyspace level - Replication Factor & Replication Startegy - Single(data replicate within same rake) & Network Topology(data replicate With different data centre)
        - create keyspace students_details with replication = {'class' : 'SimpleStrategy', 'replication_factor':1};
          use students_details;

Links

Data Model - https://www.guru99.com/cassandra-data-model-rules.html
Cassandra architecture - https://www.guru99.com/cassandra-architecture.html
Tutorial - https://www.guru99.com/cassandra-tutorial.html

Cassandra 3.11.4 path - /mnt/c/Users/vvignesw/Documents/setup/cassandra/apache-cassandra-3.11.4
Conf path - C:\Users\vvignesw\Documents\setup\cassandra\apache-cassandra-3.11.4\conf
Change the authenticator as below
authenticator: PasswordAuthenticator

Activate python 2.7 using **activate py27**
Cd C:\Users\vvignesw\Documents\setup\cassandra\apache-cassandra-3.11.4
/mnt/c/Users/vvignesw/Documents/setup/cassandra/apache-cassandra-3.11.4
bin\cqlsh

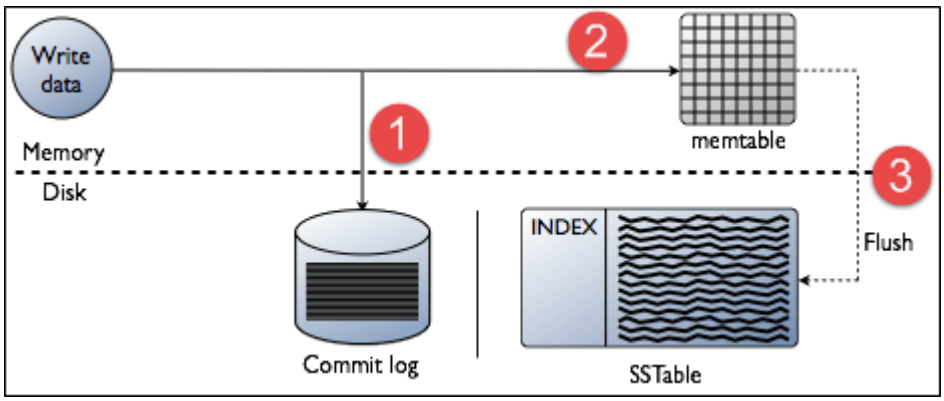
Create keyspace:
create keyspace "scene_analytics_test" with replication = {'class':'SimpleStrategy','replication_factor':1};

Go to path - /mnt/c/Users/vvignesw/Documents/java/isa/Isa-Deployment/isa-casloader/target
Create the path and file - /staging/cqlmigrate/isa/bootstrap.cql
Add the environmental property
export CAS_DATASET=/mnt/c/Users/vvignesw/Documents/java/isa/Isa-Deployment/isa-casloader/docker/dataset/isa/
export CAS_USER=cassandra
export CAS_PASSWORD=cassandra
export CAS_ADMIN_USER=cassandra
export CAS_ADMIN_PASSWORD=cassandra
export CAS_KEYSPACE=scene_analytics_test
export CAS_NODES=localhost:9042
export ISA_BUILD_VERSION=bid1.0
export CAS_REPL_FACTOR=1

export CAS_STRATEGY=SimpleStrategy

Run - java -jar isa-casloader.jar

Deactivate py 27



Caching

11 June 2019 15:06

Distributed Caching

- Components - Infinispan, Terracotta/Ehcache, Hazelcast, Memcached, Redis, Cassandra, Elasticache(by Amazon)
- **data sharding** - Sharding is a type of database partitioning that separates very large databases the into smaller, faster, more easily managed parts called data shards. The word shard means a small part of a whole
- **Cache coherent** - synching the data in with each other nodes
- **Stale data** is an artifact of caching, in which an object in the cache is **not the most recent version** committed to the data source. To avoid stale data, implement an appropriate cache locking strategy.
- **Consistent Hashing** -
    - In case of distributed caching by running caching on multiple nodes
    - the duplication is a waste of space
    - keeping the cache coherent would again be a problem
    - It wouldn't make sense to load all the data on all nodes at the same time
    - When you look for a particular key, its hash is calculated and (depending on the number of machines in the cache cluster), the cache solution knows exactly on which machine the corresponding value is located.
        - Inprocess approach - Infinispan and Hazelcast
        - own cluster of Memcached/Redis/Cassandra servers
        - Having the cache on the application nodes themselves is slightly faster than having a dedicated memory server (cluster)
    -
- Caching layer can either located on CPU memory(SRAM) - static RAM.
    - SRAM more faster than RAM
    - RAM more faster than Hard Disk(ROM)
- Caching layer can be either located **close to Application layer or Database Layer**
- Some Frameworks provide abstractions out of the box - ORMs like Hibernate
    - **2nd Level caching** by verifying in the caching first whether the data is present or ont
    - And then only it will fetch from database
- Preloading the cache if the cache data is huge
    - Batch job to fill the cache
- **Cache Configurations**
    - Cache Entries to live
    - How big do you need your cache to be(Cache maxSize)
    - Cache eviction strategy - when cache elements should expire
        - least recently used(**LRU**), least frequently used, first-in-first-out
        - Cache internally uses **double linked list** to maintain the cache eviction order
    - Types of writing data to cache
        - Write through
            - Writing to cache and database next - results in high write latency but low read latency
        - Write around cache
            - Writing to database first, bypassing the cache - results in high read latency, because fetching the data from cache result in cache-miss and it need to fetch it from database and keep it in cache
        - Write Back cache - widely used for low latency & high throughput
    - ref:https://www.computerweekly.com/feature/Write-through-write-around-write-back-Cache-explained
- Types of distributed caching configurations

    - Single node
    - Master with slave nodes
    - **Sentinel nodes**
    - Clustered nodes
    - Replicated nodes
- Cache hit - present in cache, so need to fetch it from database
- cache miss - not present in cache, need to fetch it from database

- Redis configurations & specifications - https://medium.com/@MatthewFTech/spring-boot-cache-with-redis-56026f7da83a
- https://dzone.com/articles/implementation-of-redis-in-micro-servicespring-boo
- Caching overview - https://techblog.bozho.net/distributed-cache-overview/
- Spring boot caching - https://howtodoinjava.com/spring-boot2/spring-boot-cache-example/
- Difference between in-process cache and distributed cache - https://dzone.com/articles/process-caching-vs-distributed
- https://www.javaworld.com/article/2078565/open-source-tools-use-memcached-for-java-enterprise-performance-part-1-architecture-and-setup.html

  Redis cache applications:
    - Caching
    - Session management
    - Pub-sub concept
    - Publish broadcasting
      https://github.com/redisson/redisson/wiki/7.-Distributed-collections

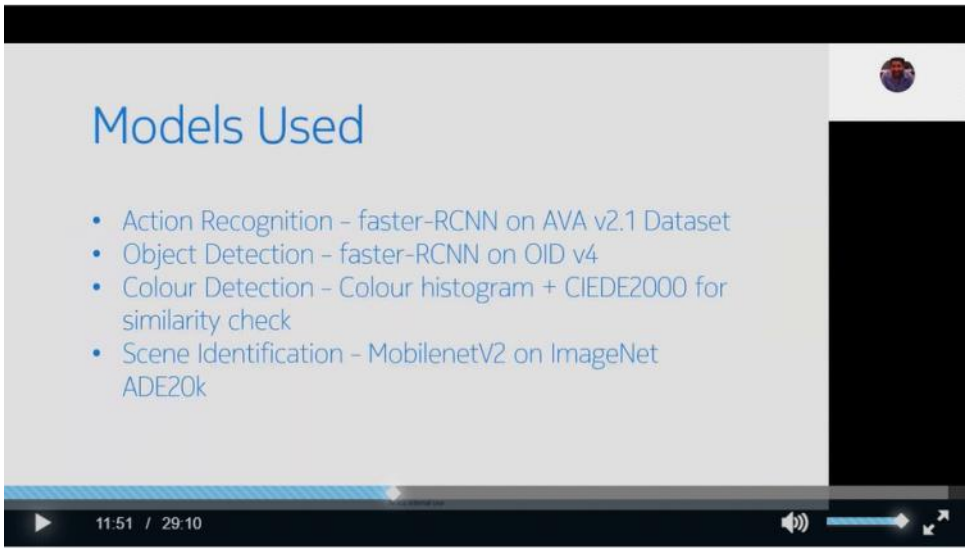
  Redis distributed caching -
  https://www.javacodegeeks.com/2019/01/spring-data-redis-high-availability-sentinel.html
  https://medium.com/@amila922/redis-sentinel-high-availability-everything-you-need-to-know-from-dev-to-prod-complete-guide-deb198e70ea6
  (topic to cover)
  https://dzone.com/articles/java-distributed-caching-in-redis
  https://medium.com/@MatthewFTech/spring-boot-cache-with-redis-56026f7da83a

  Redisson - https://www.baeldung.com/redis-redisson
  https://dzone.com/articles/distributed-java-collections-in-redis-with-redisso

Machine learning

27 June 2019 15:36

Deep Learning Algorithms



Python

09 July 2019 23:47

Diff between python 2.x and 3.x
- Range, xrange
- Map obj creates list , creates only map object

Generators - simplified form of iterators
Iterators
Decorators
Sum(iterable), next(iterable),
max(iterable, key=func()) - value returned by function will be considered to take max
String.format
List.count(no) - to get the count of particular number which we are passing
Class creation - self keyword
Inheritance - class subclass(superclass)
Sorted(iterable, key= func(), reverse=True) - based on the value returned by function it will be sorted
Functools - reduce function
Map, filter
Flatmap
Dict,List comprehension - [], {}
Zip() method - to combine individual lists into dictionaries
- {Val1:val2 For val1, val2 in Zip(key,values)}

Itertools - combinations() method - to introduce all combination of values from the list - (permutation & combination)
Extend() method - to combine two list values
Shallow copy and deep copy
Issubclass(instance, class)
Isinstance
Python support Multi inheritance - class C(A,B)
- To create subclass, call the init method of super class - A.__init__(self, name, subj)
  - super().__init__(self,name,subj)
- To call super class method - super(C, self).method() - it on MRO
- A.method(self), B.method(self)will be called based
- super().method()

MRO - Multi Resolution Order - class C(A,B)
- depth-first lookup order.
  - Depth first left to right algorithm
  - C3 linearization algorithm
- It will be called in the order of C,A,B,object

Ref: https://www.programiz.com/python-programming/multiple-inheritance
Monkey Patching - dynamically change the behaviour of the functions in the class at runtime
List methods -
- Append - (1)
- Extend - [1,2,3]
- Insert - (index, val)
- Pop()
- Remove()

obj.__dict__ - to get the variables in dictionary format

Ref: https://www.edureka.co/blog/interview-questions/python-interview-questions/
https://www.geeksforgeeks.org/minimum-number-of-adjacent-swaps-to-convert-a-string-into-its-given-anagram/
https://www.geeksforgeeks.org/print-all-possible-combinations-of-r-elements-in-a-given-array-of-

size-n/

Json - loads & jumps for converting json to python datatypes or converting python datatypes to json
Pickle - for marshalling & unmarshalling(serializing & deserializing the python data to file)
File reading & writing - open('file.txt','r'), file.read(), file.write(), w-write, a-append, r-read
Import os,

- os.listdir() - return list of string directory & files
- Os.path.join(val1, val2) - appending two strings
- Os.path.isfile(file)
- Os.path.isdir(dir)

https://www.geeksforgeeks.org/working-with-json-data-in-python/
https://www.geeksforgeeks.org/understanding-python-pickling-example/
https://www.geeksforgeeks.org/file-handling-python/

docker run -t python_print_container1 python_print_container
docker run --name python_print_container1 python_print_container

Flask & gunicorn
https://www.fullstackpython.com/blog/python-3-flask-green-unicorn-ubuntu-1604-xenial-xerus.html

Flask==**0.12.2** --version
Django (1.11, 2.0, 2.1, 2.2)

React JS

13 July 2019 17:45

Props and state
For parent-child communication, simply pass props.

Use state to store the data your current page needs in your controller-view.

Use props to pass data & event handlers down to your child components.

These lists should help guide you when working with data in your components.

Props

are immutable
which lets React do fast reference checks
are used to pass data down from your view-controller
your top level component
have better performance
use this to pass data to child components

State

Component will re-rendered
should be managed in your view-controller
your top level component
is mutable
has worse performance
should not be accessed from child components
pass it down with props instead

JSX - java script expression

```
<MyButton color="blue" shadowSize={2}>
  Click Me
</MyButton>
```

Equivalent code created by react element after compilation

```
React.createElement(
  MyButton,
  {color: 'blue', shadowSize: 2},
  'Click Me'
)
```

Babel is a free and open-source JavaScript compiler that is mainly used to convert ECMAScript 2015+ code into a backwards compatible version of JavaScript that can be run by older JavaScript engines. Babel is popular tool for using the newest features of the JavaScript programming language.

2 scenarios in small application react framework

- If the parent component has to maintain all the states of child components, then
    - Pass the event_callbacks function which perform the action of changing the state from parent to child as parameter
    - Child component will receive it as props value and child component can update the value of state of parent component with the help of callbacks
- Pass the **state** values as parameters in child component, so that child component will receive it

as **props** value

Function Components - only render method present

Object assign:
New_updated_obj = Object.assign(old_obj,{'key':'value'})
New_updated_obj = Object.assign({},{'key':'value'}) - to get the clone of the object

Object Spread: - another type of cloning
New_obj = {… old_obj}

```
const obj = { a: 'a', b: 'b', c: 'c' };
{ a: 1, b: null, c: void 0, ...obj }; // { a: 'a', b: 'b', c: 'c' }
{ a: 1, b: null, ...obj, c: void 0 }; // { a: 'a', b: 'b', c: undefined }
{ a: 1, ...obj, b: null, c: void 0 }; // { a: 'a', b: null, c: undefined }
{ ...obj, a: 1, b: null, c: void 0 }; // { a: 1, b: null, c: undefined }
```

Stateful React Component vs Stateless
Stateful - React component which carries with its own state
For smaller applications, the parent React Component can hold all the child component state.
For larger applications, redux framework helps in child components holding its own state

Advantages of redux
- Multiple react components can access the same state but don't have any parent-child relationship
- If it is difficult to pass down the state to multiple components with props

Redux Principles:
- State is immutable and cannot be changed in place
- The only way to change the state of the component is by sending the signal to the store
  Signal is nothing but the **actions**
  - **Dispatching** an action is the process of sending out the signal
  - Since the state is immutable, we cannot change the state immediately, result is copy of current data plus the new data - (new state)

Redux:
https://www.valentinog.com/blog/redux/
Reducer function -

```
function reduce(init state, actions){
    If (actions.type === 'NEW'){
        return Object.assign(old_state,{actions.value})
    }
    }
```

How to invoke the functions written in another js file?
Export default function_name;

How to import the function
Import reduce from '../reducers/index'

Actions:
Reducers will be come to know when it has to produce the next state through actions only
Actions are nothing but JavaScript Objects {'name':'vignesh,'state':'tamilnadu'}
Using dispatch() method actions has to be invoked
And it contain the value which has to be updated in new state
Function addAction(payload) {

Return **{type:'ADD', payload}**
};

Action object should always have **type** property

It is **best practice** to wrap all the action projects inside the function. Such function is **action creator**

Connecting react with redux

Npm I react-redux --save-dev

Export default connect(mapStateToProps,mapDispatchToProps)(ReactComponent)

mapStateToProps - map part of the redux states to react props

mapDispatchToProps - map the redux actions to react props so that u can call the actions directly without dispatch method

mapStateToProps

- Mainly used in listing the values present in the state thorough props
- mapStatetoProps(state) - will accept parameter as state, get the value from the state like state.articles and assign it to the key in map object
  - Eg: mapStateToProps(state) {
  - Return { articles: (state) => return state.articles};
    };
    Or
    mapStateToProps = (state) => {
    Return {articles: state:articles}'
    };

mapDispatchToProps(dispatch) {
return {addArticle: (articleObj) => dispatch(addArticle(articleObj))

handleSubmit(event) {
This.props.addArticle({this.state.title, this.stat.id})
}

https://daveceddia.com/redux-mapdispatchtoprops-object-form/

In index.js class

Import {Provider} from 'react-redux';

Import {store} from 'redux'

<Provider store={store}>
<App>
</Provider>

Creating **Store**:

Import {createStore} from 'redux';

Import { rootreducer } form 'reducers/index'

Const store = createStore(rootreducer)

Export default store;

Object destructuring - ({ someaction})

Const component = ({someAction}) = {someaction()};

https://stackoverflow.com/questions/42871136/dispatch-function-in-redux-react

docker

17 July 2019 16:03

Difference between CMD and ENTRYPOINT - https://goinbigdata.com/docker-run-vs-cmd-vs-entrypoint/

Spark streaming

18 July 2019 21:55

Install spark and pyspark using findspark
https://medium.com/@naomi.fridman/install-pyspark-to-run-on-jupyter-notebook-on-windows-4ec2009de21f
https://bigdata-madesimple.com/guide-to-install-spark-and-use-pyspark-from-jupyter-in-windows/

Conda-findspark package installation
Command: conda install -c conda-forge findspark

Mongodb docker container creation:
https://www.thepolyglotdeveloper.com/2019/01/getting-started-mongodb-docker-container-deployment/

https://blog.unnati.xyz/mongodb-and-pyspark-2-1-0-431bab0b5465

Spark query building
https://gist.github.com/raghothams/6183df0e60ed62989be448795360d27b?source=post_page----------

Kafka streaming with spark
https://www.rittmanmead.com/blog/2017/01/getting-started-with-spark-streaming-with-python-and-kafka/
Github code: https://gist.github.com/rmoff/eadf82da8a0cd506c6c4a19ebd18037e
https://spark.apache.org/docs/latest/streaming-programming-guide.html
Foreach & foreachRDD - https://github.com/danielsan/Spark-Streaming-Examples/blob/master/spark-streaming-foreachRDD-and-foreach.py

Kafka bitnami cluster distribution
https://github.com/bitnami/bitnami-docker-kafka

Kafka topics creation/listener through command line
http://cloudurable.com/blog/kafka-tutorial-kafka-from-command-line/index.html

```
kafka/bin/kafka-topics.sh --create   --zookeeper zookeeper:2181   --replication-factor 1 --partitions 1   --topic mytopic
```

kafka-console-producer.sh --broker-list localhost:9092 --topic mytopic

```
kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic mytopic --from-beginning
```

kafka-topics.sh --describe --zookeeper zookeeper:2181 --topic mytopic

Spark streaming with kafka, spark and mongodb(scala)
https://rklicksolutions.wordpress.com/2017/04/04/read-data-from-kafka-stream-and-store-it-in-to-mongodb/

Spark Link - https://www.whizlabs.com/blog/do-you-need-hadoop-to-run-spark/
https://databricks.com/blog/2014/01/21/spark-and-hadoop.html

Spark
- SparkConfig
  - Configure the master node URL
  - Configure the app name
  - Configure the key-value pairs(basically the properties)
- SparkContext

Serializer - incae of kafka what kind of in/out serialized i.e., string, int serializers
Master
AppName

Spark uses Py4j to launch the JVM and creates the JavaSparkContext
Run Spark Appl -> driver Program starts -> It has (Main func()) -> SparkContext intiation
Driver program -> runs the operation inside executors on worker nodes

3 ways to run spark on hadoop cluster

1. Standalone
2. Over YARN(Hadoop 2.x)
3. In MapReduce(SIMR)

Spark Ecosystem
1. Spark Core - SparkConfig, SparkContext, SparkSession
2. Spark Streaming - StreamingContext
3. Spark SQL - SqlContext
4. Spark Cassandra Connector
5. Graph Analytics
6. Machine Learning Library

RDD

Resilent Distributed Dataset
immutable
It is fault-tolerant
Computations in RDD are automatically parallelized across cluster
To apply operations on RDD, you need 2 ways,
Transformation
Action
Collect() method - to bring all the distributed data into the single node
So its preferable to use take(2) to limit the result set always to avoid out of memory
Ref: https://www.tutorialspoint.com/pyspark/pyspark_rdd.htm
https://www.edureka.co/blog/pyspark-rdd/

Map Vs FlatMap -
Map - Transforming each element into another element, so length of both array will be same
FlatMap - Transforming each elements into multiple elements, so length of both array will be different
Ref : https://data-flair.training/blogs/apache-spark-map-vs-flatmap/

GroupByKey -
Grouping the elements based on the key - eg.[('vig',1),('vig',2),('ven',1),('ven',2)] ->
[('vig',[1,2]), ('ven',[1,2])]
https://backtobazics.com/big-data/spark/apache-spark-groupbykey-example/

Groupby -
Doing grouping based on some function. Eg.based on length

SQLContext
To create DataFrames

SparkSession vs SparkContext
https://data-flair.training/forums/topic/sparksession-vs-sparkcontext-in-apache-spark/

Pip install findspark

Mongodb and kafka streaming related dependencies:
org.mongodb.spark:mongo-spark-connector_2.11:2.4.0,
org.apache.spark:spark-streaming-kafka-0-8_2.11:2.4.3

Mongo commands
Mongodb image: mongo:latest
docker exec -it 828978fbf59f bash
Execute mongo command as bash:#? **Mongo**
Inside mongodb shell:

show dbs;	To list out the databases
Db;	to connect to the specific database
use kafkadata;	To connect to the specific database
db.kafkadata.insertOne({'firstName':'vignesh','lastName':'venkatesh'})	
db.kafkadata.find({'firstName':'vignesh'})	Filtering out based on the fields
show collections;	To show the list of collections(tables) from the connected databases;
db.people.find();	
db.kafkadata.insertMany({'firstName':'vignesh','lastName':'venkatesh'})	

Kafka Steaming
Kafka Images : bitnami/kafka:latest, bitnami/zookeeper:latest
Path : C:\Users\vvignesw\Documents\java\kafka, docker-compose3.yml
Kafka distribution: Bitnami/kafka

Path for shell files: /opt/bitnami/kafka/bin/

kafka-console-producer.sh --broker-list localhost:9092 --topic mytopic
kafka-topics.sh --create --zookeeper zookeeper:2181 --replication-factor 1 --partitions 1 --topic mytopic
kafka-topics.sh --describe --zookeeper zookeeper:2181 --topic mytopic

Path: python scripts\kafkaProducer.py, kafkaConsumer.py

Kafka-python package : https://towardsdatascience.com/kafka-python-explained-in-10-lines-of-code-800e3e07dad1
Docs & code : https://kafka-python.readthedocs.io/en/master/usage.html#

sparkML - https://www.datacamp.com/community/tutorials/apache-spark-tutorial-machine-learning?utm_source=adwords_ppc&utm_campaignid=1455363063&utm_adgroupid=65083631748&utm_device=c&utm_keyword=&utm_matchtype=b&utm_network=g&utm_adpostion=1t1&utm_creative=278443377092&utm_targetid=aud-390929969673:dsa-473406582395&utm_loc_interest_ms=&utm_loc_physical_ms=9061999&gclid=CjwKCAjw-ITqBRB7EiwAZ1c5U7nAy6kb-Ze_KU-0wK3s-dktEancpfAgPfwuRUbjn4ul2hYZ5Ys4lBoCcNgQAvD_BwE

Connecting to postgresql SQL from spark - https://medium.com/@usmanazhar4/how-to-read-and-write-from-database-in-spark-using-pyspark-150d39cdbb72

https://spark.apache.org/docs/latest/sql-data-sources-jdbc.html

Kafka

25 July 2019 21:20

Kafka cluster - multiple nodes running on same network
Kafka nodes runs on top of zookeeper, so zookeeper knows how many nodes are connected to it.
Kafka topics - topics are **queues** basically
Kafka partitions -
splitting the topics(queues) across the nodes to handle the streaming data load from the producers
Data will be distributed across the nodes in round-robin fashion with replication factors
Whichever nodes receive the data will act as the master and replicate the data to other nodes using zookeeper

Command to create the kafka topics in a cluster -
Bin/kafka/kafka-topics.sh --create --zookeeper localhost:9011 --replication-factor 2 --partitions 2 --topic mytopic
Bin/kafka/kafka-topics.sh --describe --zookeeper localhost:9011 --topic mytopic

Kafka-producer API
Kafka-consumer API
Kafka-streams API
Kafka-connectors API

@KafkaListener(topicPartitions = @TopicPartition(topic="mytopic", partitions={"1"}, group="sensorGroup"))
Public void methodname(@Payload String msg, @Headers(KafkaHeaders.RECIEVED_PARTITION_ID))

Properties can be done using 2 ways
1.Application.yml
2.@configuration
ProducerFactory
KafkaTemplate

Application.yml

```
spring:
  kafka:
    consumer:
      bootstrap-servers: localhost:9092,localhost:9093 -- list of kafka servers
      group-id: jsa-group, sensor-group --list of consumer groups
      auto-offset-reset: earliest
      key-deserializer: org.apache.kafka.common.serialization.StringDeserializer
      value-deserializer: org.apache.kafka.common.serialization.StringDeserializer
    producer:
      bootstrap-servers: localhost:9092,localhost:9093
      key-deserializer: org.apache.kafka.common.serialization.StringDeserializer
      value-deserializer: org.apache.kafka.common.serialization.StringDeserializer
```

ProducerFactory - (ProducerConfig.java)
Bootstrap-servers-config
Key-serializer
Value-serializer

KafkaTemplate(producerFactory object)

Producer

@autowired
Kafkatemplate<String,String> template

SendResult<String, String> result =Template.send(topic, data).get()
RecordMetaData metadata = Result.getMetaData()
metaData.partition(), metaData.offset()

https://content.pivotal.io/blog/understanding-when-to-use-rabbitmq-or-apache-kafka

Django Web App

04 September 2019 23:26

Django project consist of multiple applications running independently.

Each application follows Model-View-Template pattern

A Product may consists of more than 1 features. Each feature can be created like a application

This structure helps in moving the applications faster among different projects

Django commands

Create Project

Django-admin startproject myproject

Myproject/

Manage.py

Myproject/

__init__.py

Url.py

Wsgi.py

Settings.py - debug, database configuration, installed_Apps, middleware_classes

Settings.py

DEBUG = true

Databases = {'ENGINE':'', 'NAME':''}

MIDDLEWARE_CLASSES = ('django.contrib.sessions.middleware.SessionMiddleware','CsrjViewMiddleware','Authenticate Middleware',)

Apps LifeCycle

Python manage.py startapp myapp

Myapp/

__init__.py

Views.py - controller

Models.py - model bean

Tests.py - to write unit test cases

Admin.py - to make app configurable to admin interface

Admin Interface

Python manage.py migrate - to create the necessary tables

Syncdb will create tables based on the datatype we defined in model classes

Python manage.py createsuperuser - to create superusers

Myproject/url.py

From django.conf.urls import patterns, url, include

From django.contrib import admin

Urlpatterns = patterns('', url(r^'admin/', include(admin.site.urls))

Add the created app in the **installed_apps** list

Settings.py
INSTALLED_APPS = ('django.contrib.admin', '**myapp**')

Python manage.py runserver - to start the server

Django - security
https://docs.djangoproject.com/en/2.2/topics/security/

Creating Models

Models.CharField('name',max_length=30)
Models.ForeignKey(Blog, on_delete= models.CASCADE)
Models.ManyToManyField(Author)
Models.IntegerField('name')
Models.FileField
Models.ImageField(upload_to='/dir')
In settings.py, MEDIA_ROOT = ''

B = blog(name='vignesh', tagline = 'something')
b.save() - to save the existing data
b.create() - to create it freshly and save

Entry.objects.get(pk=1)
Entry.objects.get(name='vignesh')
Entry.objects.all()

ManytoMany
Entry.authors.add(johin, paul, george)

Storing cookies:

Reponse.set_cookies('last_connection',2)
Response.set_cookies('username','vignesh')

If 'username' in request.COOKIES:
Username = Request.COOKIES['username']
https://www.tutorialspoint.com/django/django_cookies_handling.htm

Saving sessions:
'django.contrib.sessions' - installed_apps
'django.contrib.sessions.middleware.SessionMiddleware' - middleware_classes

Django can save the session info in **database**(django_session table)
It can be stored in other ways like **file**, or in **cache**

When session enabled, request will have the session attribute

Views.py
Def formview(request):
If request.session.has_key('username'):
Username = request.session['username']
Else:
Render(request, 'login.html', {})

Def login(request):

If request.method == 'POST':

Request.session['username'] = username

Url.py
Urlpatterns = urlpattern('myapp.views', url(r^'/connection', 'formView', name='loginform'),
Url('r^/login', 'login', name='login'))

Views.py
Def logout(request):

Del request.session['username']

Url.py

Urlpatterns = urlpattern('', Url('/logout', 'logout', name='logout'))

Django_caching

Middleware configuration
Database based caching
In settings.py file
CACHES = { default: {'backend': 'databaseCache', 'LOCATION':'tablename'}}
Python manage.py createcachetable - to create caching table

File based caching
CACHES = { default: {'backend': 'FileBasedCache', 'LOCATION':'/var/tmp/django_cache'}}

Caching the entire site - by configuring the middleware

Middleware_classes += ('UpdateCacheMiddleware,
'CommonMiddleware', 'FetchFromCacheMiddleware') - order important

CACHE_MIDDLEWARE_ALIAS
CACHE_MIDDLEWARE_SECOUNDS - no of secs

Caching the particular view

@cache_page(60 *15) - 15 mins

Setting up the cache in Memory

Setting up views(Controller):

Getting Values from HttpRequest:

Request.GET
Request.POST
Request.FILES
Request.META - dictionary like object which holds value like content_length, content_type, server_name etc..
Request.COOKIES

Attributes set by middleware in httprequest:

AuthenticationMiddleware:

If the user logged-in properly, then request.is_authenticated will be true, else false

CurrentSiteMiddleware:
Instance of site returned by get_current_site() from request object about current_site.

SessionMiddleware:
Dictionary like object representing current session. Request.session

Setting up responses:
From django.http import JsonResponse

Response = JsonResponse({'foo':'bar'})
Return response

StreamingHttpReponse - for generating large csv files

FileResponse - subclass of StreamingHttpResponse

Django RestFramework

Pip install djangorestframework

Django-admin startproject tutorial
Cd tutorial
Django-admin startapp quickstart
Cd ..

Settings.py
Installed_apps = ['', 'rest_framework']

Pagination

Serializers
https://www.django-rest-framework.org/tutorial/1-serialization/
From rest_framework import serializers

Class snipperSerializer(serializers.Serializer):

Id = serializers.IntegerField(read_only=true)
Title = serializers.CharField(max_length=100)

Def create(self, validated_data):

Def update(self, validated_data):

To simplify the code present in the serializers class, django comes up with ModelSerializers

Class snippetSerializer(serializers.ModelSerializer):

Model = Snippet
Fields = ['name,'col',]

Resframework.renderers import JSONRenderer
Restframework.parsers import JSONParser

Model to Python Speicfic Seriazlied One:

Serializer = SnippetSerializer(model)
Serializer.data - python specific datatype

Deserialization to convert Python Data type to Model:
Serializer = SnippetSerializer(data = val)
Serializer.is_valid() - True

Serializer.validated_data - OrderedDict
Serializer.save() - Snippet Model Object saving option
Serializer.data - python specific datatype

Serializer = SnippetSerializer(model.objects.all(), many = True)
Serializer.data - OrderedDict - model instance to python specific datatype

Views
https://www.django-rest-framework.org/tutorial/3-class-based-views/

Function based views
Class based views - latest

ModelViewSet

APIView

```
Def get(self, request,  pk, format=None):
        Snippet = self.get_object(pk)
        Serializer = SnippetSerializer(snippet)
        Return reponse(Serializer.data)

Def put(self, request, pk, format=None):
        Snippet = Self.get_object(pk)
        Serializer = SnipperSerializer(snippet, data =request.data)
        If serializer.is_valid():
                Serializer.save()
                Return response(serializer.data)
        Return reponse(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
Def get_object(self, pk):
        Try:
                Return Snippet.objects.get(pk = pk)
        Except snippet.DoesNotExist:
                Raise Http404
```

Urls.py

```
From django.urls import path
From rest_framework.urlpatterns import format_suffix_patterns

Urlpatterns = [
        Path('snippets/', views.SnippetList.as_view()), path('snippets/<int:pk>',
        views.snippetdetail.as_view())]
```

ListCreateAPIView - (list, create)

```
Queryset = snippets.objects.all()
Serializer_class = Snippetserializer
```

RetrieveUpdateDestroyAPIView - (get, update, delete)

```
Queryset = snippets.objects.all()
Serializer_class = Snippetserializer
```

Jenkins

19 September 2019 14:59

https://www.blazemeter.com/blog/how-to-use-the-jenkins-declarative-pipeline/

https://jenkins.io/doc/book/pipeline/syntax/

Kerberos

GENERAL QUESTION

21 September 2019 21:14

sortMapBasedOnValues - sortMapBasedOnValues.java
Evict() method in hibernate
Prevent singleton pattern - https://www.geeksforgeeks.org/prevent-singleton-pattern-reflection-serialization-cloning/
    Breaking the singleton using
        Cloning - throw cloningNotSupportedException, Cloneable
        Reflection
        Serialization - implement readResolve() method

How to compare the 2d array in java whether it is equal or not in time complexity of O(n)
Is it possible to use where clause and having clause in single sql query -
    having clause have to be used in place of aggregate functions to be used along with GroupBy clause
Strategy Design Pattern
2 level comparison in comparator using Comparator Interface
    In Compare() method, you have to use **string.compareTo()** to compare 2 string to return int values like
        0 - if it is equal
        -1,+1 - based on which values is bigger
    If (namecompare == 0) {
        Return (agecompare==0)?namecompare:agecompare;
    } else {
        Return namecompare;
    }

Collections.sort() internally uses Strategy Design Pattern
@ControllerAdvice,

UserDefinedException - refer https://www.geeksforgeeks.org/throwable-initcause-method-in-java-with-examples/

BeanPostProcessor - postProcessBeforeInitialization, postProcessAfterInitialization
@EventListener(ApplicationReadyEvent.class)

Application Context - event publishing
1. Event should **extend** ApplicationEvent
2. Publisher should inject ApplicationEventPublisher object
3. Listener should **implement** ApplicationListener interface

Scene Analytics setup

07 March 2019 12:06

Redis start - C:\Users\vvignesw\Documents\setup\redis_2.4.5\64bit

MariaDB connect in dev

02 July 2019 11:48

Connect to the mariadb pod using kubectl command
kubectl exec -it isa-csfmariadb-mariadb-0 bash

isa-csfmariadb-mariadb-0 - master node
isa-csfmariadb-mariadb-1 - slave node

Execute the Mysql command to connect to database:
mysql -u iotadmin -pmfrug

S.no	Sql commands	description
	use isanode	to connect to particular database **isanode**
	show tables;	to show the list of tables
	describe white_label;	To describe the structure of the table
	select count(1) from white_label;	
	select WHITE_LABEL_ID, group_id, logo_img_name, backgrnd_img_name, settings, default_logo from white_label;	

Get gateways statistics

07 March 2019 12:38

Gateway Details like Last Reboot, Backup Details, Component Versions etc

gatewayController

/gateway/status

ReposneData:
onlineCamCount,offlineCamCount,lastReboot,diskTotal,diskUsed,diskAvailable
memoryTotal,memoryUsed,memoryAvailable

git commit -m "AAIOTCSA-2304:get gateway details with backup, versions from prop as well"

AAIOTCSA-2305

11 March 2019 14:46

GatewayServiceImpl.java - getGatewayDetails
GatewayDaoImpl - getGatewayPropValue

git commit -m "AAIOTCSA-2305:get gateway statistics details - added audit logging info code "

Store label images in mariadb

14 June 2019 14:16

Tenant management application

Pre requistes to start the server

MariaDB -> C:\mariadb\heidisql.exe
Username:root
Password:mfrug

KeyCloak -> C:\Users\vvignesw\Documents\setup\keycloak-4.1.0.Final

To start the tenant management application

To find the target war file
C:\Users\vvignesw\Documents\java\management1\managementserver\target

Mvn clean
Mvn install
Mvn spring-boot:run

Unix commands

04 April 2019 13:03

Basic linux commands
https://www.guru99.com/must-know-linux-commands.html

Command	Description
ls	Lists all files and directories in the present working directory
ls - R	Lists files in sub-directories as well
ls - a	Lists hidden files as well
ls - al	Lists files and directories with detailed information like permissions, size, owner, etc.
cat > filename	Creates a new file
cat filename	Displays the file content
cat file file2 > file3	Joins two files (file1, file2) and stores the output in a new file (file3)
mv file "new file path"	Moves the files to the new location
mv filename new_file_name	Renames the file to a new filename
sudo	Allows regular users to run programs with the security privileges of the superuser or root
rm filename	Deletes a file
man	Gives help information on a command
history	Gives a list of all past commands typed in the current terminal session
clear	Clears the terminal
mkdir directoryname	Creates a new directory in the present working directory or a at the specified path
rmdir	Deletes a directory
mv	Renames a directory
pr -x	Divides the file into x columns
pr -h	Assigns a header to the file
pr -n	Denotes the file with Line Numbers
lp -nc lpr c	Prints "c" copies of the File
lp -d lp -P	Specifies name of the printer
apt-get	Command used to install and update packages
mail -s 'subject' -c 'cc-	Command to send email

address' -b 'bcc-address' 'to-address'	
mail -s "Subject" to-address < Filename	Command to send email with attachment

Symbols - https://superuser.com/questions/247127/what-is-and-in-linux
Function calling & args - https://bash.cyberciti.biz/guide/Pass_arguments_into_a_function
https://unix.stackexchange.com/questions/109625/shell-scripting-z-and-n-options-with-if

Command substitution -
https://www.tldp.org/LDP/abs/html/commandsub.html#COMMANDSUBREF
Eg: Var=`ls *.txt`
Basic linux commands - https://www.guru99.com/must-know-linux-commands.html#17
String comparison - https://bash.cyberciti.biz/guide/String_comparison
If-else statement - https://www.tutorialspoint.com/unix/if-elif-statement.htm

	https://tecadmin.net/tutorial/bash/examples/check-if-string-contains-another-string/

Value substitutions - https://www.tutorialspoint.com/unix/unix-shell-substitutions

Sed, awk & grep commands to deal with file, iterating through the lines in the files

`$?`-The exit status of the last command executed.
`$0`-The filename of the current script.
`$#`-The number of arguments supplied to a script.
`$$`-The process number of the current shell. For shell scripts, this is the process ID under which they are executing.

```
$ echo "$TEST"
test
$ echo '$TEST'
$TEST
```

Chmod understanding - https://askubuntu.com/questions/932713/what-is-the-difference-between-chmod-x-and-chmod-755/932860

Chmod +x = ugo+x = u+x, g+x, o+x
+ means addition of rules in addition to existing
= means clearing old rules and adding the current rule
U - current user(owner), g - group , o - others
755 -
7 - 111(binary) - U(4) - rwx - read, write, execute
5 - 101(binary) - g(5) rx - read, execute
5 - 101(binary) - O(5) rx - read, execute
421 - (100, 010, 001) - (r--, -w-, --x)

Concurrent Package

30 June 2019 13:09

Rentrant Lock -

- Lock.tryLock(long timeout, TimeUnit.Seconds) - threads to time-out instead of waiting indefinitely
- When more than one thread are competing for the same lock, the lock favours granting access to the longest waiting thread
- Private final RentrantLock lock = new RentrantLock();
    - Lock.tryLock(fime, TimeUnit.Seconds)
    - Lock.lock(); - before accessing the linkedlist
    - Lock.unlock(); - once the operation got completed in linkedlist

https://www.geeksforgeeks.org/reentrant-lock-java/
https://examples.javacodegeeks.com/core-java/util/concurrent/locks-concurrent/reentrantlock/java-reentrantlock-example/

Java 8 - Nashorn JavaScript(JavaScript Engine)

Countdown latch

Latch.countdownlatch(5); - initialize with no of threads for countdown

Latch.countdown() - aftercompleting the task, execute the countdown

Latch.await() - in main thread - main thread will wait for countdown to come to zero.

Once It came, it will execute the remaining task

Link - https://www.geeksforgeeks.org/countdownlatch-in-java/

Cyclic Barrier

Final cyclicbarrier = new cyclicbarrier(3, new Runnable() {

@override

Public void run() {

System.out.println("all the threads are arrived at the barrier, so we can get the total count of north, south & west regions");

}}

Barrier.await() - In each thread, after all the process are completed, it has to invoke await()

Link - https://javarevisited.blogspot.com/2012/07/cyclicbarrier-example-java-5-concurrency-tutorial.html

ExecutorService

Types

newSingleThreadExecutor

NewFixedThreadPool(no of Threads)

newScheduledThreadPool(no of Threads)

Different ways to delegate tasks

Execute(Runnable) - accepts Runnable and return void

Submit(Runnable) - accepts Runnable and return Future Object

Future Object will be used here to verify whether the process executed successfully or not

If it returns null, then the process executed successfully

Future<Object> Submit(Callable) - returns future which can help us in getting the values using get() method

Future.get()

Executor.awaitTermination(5, Time.SECONDS)

Executor.shutdown()

invokeAny(List<Callable> listOfCallables) - will invoke any one of the callables in list and return Future object which belongs to any of the Callable

invokeAll(List<Callable> listOfCallables) - will invoke all the callables in list and return List<Future>

Callable and Runnable Difference

```
Class Printer implements Runnable {
    Public void run() {
    }
}
Class Printer implements Callable {
    Public Object call() throws Exception{
    }
}
```

Spring boot

17 July 2019 14:39

Spring cloud frameworks
https://www.javainuse.com/spring/spring-cloud-interview-questions

Spring boot with microservices using kubernetes

https://www.baeldung.com/spring-boot
Spring boot start - how to build a simple app - https://www.baeldung.com/spring-boot-start

@configurationproperties - https://www.baeldung.com/configuration-properties-in-spring-boot
To load the application.properties entries into the beans

Feign client - advanced rest client - abstraction over REST-based calls
Feign internally uses Ribbon for client-side load balancing
Feign as client - https://dzone.com/articles/microservices-communication-feign-as-rest-client
https://www.javainuse.com/spring/spring-cloud-netflix-feign-tutorial

Zuul and Ribbon integration
https://stackoverflow.com/questions/43538030/zuul-and-ribbon-integration

Zuul - Gateway for reverse proxy, load balancing - **server side load balancer** - (internally uses ribbon)
Ribbon - Client Side load balancing

- Ribbon fetches the list of pods/instances running for that particular services from the service registry
- It finds the which pod/instances has more bandwidth to accept the request

Kubernetes dependency

```
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-kubernetes</artifactId>
<version>0.3.0.BUILD-SNAPSHOT</version>
</dependency>
```

Service Discovery
Service can be used to access the application outside the Kubernetes cluster or for inter-service communication inside a cluster.

```
apiVersion: v1
kind: Service
metadata:
  name: employee
  labels:
    app: employee
spec:
  ports:
  - port: 8080
    protocol: TCP
  selector:
    app: employee
```

@EnableDiscoveryClient - has to be enabled in the started file of spring boot application

Add entry in the application.properties

- Give the same name of the service as the application name

```
spring:
  application:
    name: employee
```

Ribbon & feign client for inter service communication

```
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-kubernetes-ribbon</artifactId>
<version>0.3.0.BUILD-SNAPSHOT</version>
</dependency>
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-openfeign</artifactId>
</dependency>
```

Microservices using kuberentes - https://dzone.com/articles/quick-guide-to-microservices-with-kubernetes-sprin

@EnableDiscoveryClient
@EnableFeignClients
@EnableMongoRepositories
@EnableCassandraProperties(basePkg = "org.com.*")

Creating ingress as gateway to expose these services to outside world -

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
 name: gateway-ingress
 annotations:
  nginx.ingress.kubernetes.io/rewrite-target: /
spec:
 backend:
  serviceName: default-http-backend
  servicePort: 80
 rules:
 - host: microservices.info - context name to be called from external APIs
  http:
   port: 8080 -- to invoked from the external APIs
   paths:
   - path: /employee
     backend:
       serviceName: employee
       servicePort: 8080
   - path: /department
     backend:
       serviceName: department
       servicePort: 8080
   - path: /organization
     backend:
       serviceName: organization
       servicePort: 8080
```

Annotations:

@SpringBootApplication - @Configuration, @EnableAutoConfiguration, @ComponentScan

@PropertySource(value="file:/path/application.properties") - to locate property sources from external path, not from internal path of war file

@EnableScheduling - Cron schedulers
@Scheduled(cron = "10 10 20 * * *") - Cron expressions

@EnableAsync - threadpoolexecutor

@Async("jobExecuter")
Func method

@Bean(name = "jobExecuter")
public ThreadPoolExecutor jobExecutor()

@**EventListener**(ApplicationReadyEvent.class) - triggered when the application starts

@Configuration
@Bean - to create the beans and get managed by spring

To load the property file entries into the spring beans
@PropertySource("classpath:app.properties")
@ConfigurationProperties(prefix="mail") - java config file
@EnableConfigurationProperties(configprop.class)

@Value("${MANAGEMENT_DATASOURCE_USERNAME}") - to load the value from property

@ComponentScan(basePackages = "com.nokia.managementserver") - to scan all the @Component based files

@EntityScan(basePackages = { "com.nokia.managementserver" }) - to scan the entities

@EnableCaching

@requestmapping(value='contextpath/name', method="post", header='application/xml', consumes='multipart/form-data', produces='application/json')

Link - https://stackoverflow.com/questions/30923249/spring-4-requestmapping-consumes-vs-headers

@requestmapping(value='/path/{grpId}', consumes='application/json', produces='application/json', method=
post") {
Public void addData(@pathVariable(value="grpId") String grpId, @RequestBody Group grp) {}

@Postmapping
@Getmapping

@requestPart MulipartFile file, @requestPart String id - incase if the api consumes multipart form-data, then we have to use requestPart
'multipart/form-data' request - receive the data using @requestParam , @requestPart

/context/{id}?data=val - to receive the query params
@requesmapping(value="/context/{id}", produces="application/json", method="GET") {
Public String getData(@pathVariable String id, @RequestParam("data") String val) {
Return gson.toJson(Bean Obj);

Gson.toJson(Bean Obj) - to convert the bean object to json
Gson.fromJson(json_val_string, Staff.class) - to convert the stringified json to Bean Object

@requestbody BeanObject - to send bean as part of request body in the form of JSON/XML

@EnableJpaRepositories(entitymanagerFactoryRef="", transactionMangerRef="", basePackages = {org.package}) - add this annotation in springbootmainapplicationclass

@Primary
https://www.baeldung.com/spring-primary
@Component
Interface implements 2 sub-class
Add one annotation as @primary

@Autowire this component in another bean using Dependency Injection(DI)
Primary related bean will get injected dynamically

@Autowired - by name, by type, by constructor
@Qualifier("man")

ApplicationContext vs Bean Factory difference

https://javarevisited.blogspot.com/2012/11/difference-between-beanfactory-vs-applicationcontext-spring-framework.html

ApplicationContext	Bean Factory
getBean	getBean
Creates singleton bean for all Registered beans	Wont create any beans until the invocation of getBean("na")
ApplicationContext implements BeanFactory	
Adopts internationalization	
Beans which are registered as listener Ability to publish events to listener beans	

https://howtodoinjava.com/spring-boot2/spring-retry-module/
https://www.baeldung.com/spring-retry
Spring Retry
@EnableRetry - in root of the application
@Retryable - in each methods
@Recover

Hystrix - circuit breaker
https://howtodoinjava.com/spring-cloud/spring-hystrix-circuit-breaker-tutorial/
https://spring.io/blog/2019/04/16/introducing-spring-cloud-circuit-breaker
https://www.javainuse.com/spring/spring_hystrix
https://javainuse.com/spring/spring_hystrix_circuitbreaker

Add dependency
Org.springframework.cloud
Spring-cloud-starter-hystrix

Add annotations in root application
@EnableCircuitBreaker

Add annotation in controller on each controller method
@HystrixCommand(fallbackmethod= "getdatafallback")

Public Employee getDataFallback() {}

DatasourceConfig Class
@EnableTransactionManagement
@ComponentScan

Creating Datasource

Public DataSource createDatasource() {

Datasource.url:url
Username:name
Password:pass
Max_pool_size:100
Config = new hikariConfig(props)
Return new hikariDatasource(config)
}

Creating jdbctemplate, namedjdbctemplate

Public jdbctemplate getjdbcTemplate() {
    Temp = new jdbcTemplate(new Datasource());
Namedjdbctemplate() method

Creating entityManager
LocalEntityManagerFactoryBean em = new LocalEntityManagerFactoryBean ();
Em.setDataSource(datasource)
Em.setPackagesToScan(new String[] {"com.nokia"})
Em.setJpaVendorAdaptor()
Entitymanager properties:
    Hibernete dialect - mysqlDialect - dialect represents which database to be used for hibernete
    Connection_pool_size
    Use_query_cache:true
    Second_level_cache:true
    Jdbc_batch_size:100
    Order_inserts:true
    Order_updates:true

Em.setJpaPropertyMap(properties)

Creating transactionManager
PlatformTransactionManager
Jpatransactionmanager manager = new JpaTransactionManager();
Manager.setEntityManagerFactory(entityManagerFactory().getObject());

Steps for datasource configuration
Note: add enabletransactionmanagement annotation in config class

1. Creating datasource by passing database credentials
2. Create jdbc template
3. Create entity manager by passing dialect-mysql,postgresql dialects, datasource ref, second level cache, query cache, batch sizw
4. Create transaction manager by passing entity manager

Creating cachemanager
Ehcachemanager(new ClassPathResource(ecache.xml)

CassandraConfiguration

@Table
@Column
@PrimaryKeyColumn(name="", ordinal=0, type=PrimaryKeyType.Partitioned/Clustered)

CasandraConfig.java
@EnableCassandraProperties(basePackages="org.nokia")
Creating session
Inputs:
Cluster

Creating cassandratemplate based on session

Link - https://www.baeldung.com/spring-data-cassandra-tutorial

SpringBootMainApplication
@springbootapplication - @componentscan, @enableautoconfiguration

Public static void main(String args[]) {
SpringApplication.run(springmainapplication.class, args);

Entity

@Entity
@Table(name="group")

@Id
@SequenceGenerator(name, sequenceName)
@GeneratedValue(strategy = AUTO, generator="sequenceGenerator")
@Column(name="id")

@Column(name="std", unique=true, nullable=false)
@Transient

@Repository
@Transactional
Interface groupRepository implements CrudRepository<Group, String> {

@Query("Select * from data where col1=:qid and col2=:qid2")
List<Group> findbyGroupIdAndData(@Param("qid") String qid, @Param("qid2") String qid2);

How to resolve LazyInitializationException
1) Hibernate.initialize(user.getCreatedJobs())
2) @OnetoMany(mappedby="creator")
    a. To @OneToMany(fetch = FetchType.EAGER, mappedby="creator")
3) JOIN FETCH
4) DTO Projection

FetchType.EAGER, FetchType.LAZY
https://vladmihalcea.com/the-best-way-to-handle-the-lazyinitializationexception/

backing services
https://www.nginx.com/blog/microservices-reference-architecture-nginx-twelve-factor-app/
https://12factor.net/processes

https://spring.io/blog/2015/01/27/12-factor-app-style-backing-services-with-spring-and-cloud-foundry

12 factors for microservices

1) One codebase per services,
2) Explicitly declare and isolate dependencies between services by binding dependency manager with the os platform/distribution itself
3) Store the configuration as the environmental variable
4) Treat backing services as attached resources, because it should be loosely coupled and provide more flexibility to the developers
5) Strictly separate build, run stages
6) Execute the app in one or more **stateless** instances and share the stateful data between instances using backing services
7) Each service manage its own data
8) Each services can be scaled out independently
9) Maximize **robustness** by making instances of services **disposable** with fast startup, shutdown. This can be easily achieved using docker containers. We can store the session data in backing services incase the container went down
10) Treat logs as event streams
11) Run admin and management task as one-off process
12) Keep development, staging & production as similar as possible to reduce the bugs that can arised in different environments

Rolling deployment, canary deployment
https://www.youtube.com/watch?v=YQLOcjvbo9s

Istio- service mesh
https://www.digitalocean.com/community/tutorials/how-to-install-and-use-istio-with-kubernetes

Spring cloud sleuth- for logging
https://spring.io/projects/spring-cloud-sleuth
https://dzone.com/articles/monitoring-microservices-with-spring-cloud-sleuth

Java 8 Features

20 July 2019 01:13

Heap space vs stack
https://www.journaldev.com/4098/java-heap-space-vs-stack-memory
Hash code and equals contract
https://www.baeldung.com/java-equals-hashcode-contracts

Java 8 interview questions
https://www.baeldung.com/java-8-interview-questions

Streaming API - map, filter & reduce
List.stream().map(data -> data*data).collect(Collectors.toList())
List.stream().map(data -> data*data).foreach((v) -> System.out.println(v))

Transformation function(**Intermediate**) - map, filter, reduce
Action functions(**Terminal**) - toArray(), collect(Collectors.toList()), foreach()

Files for ref:flatMapcheck, SampleTest, flatMapCheck1.java

SplitIterator -- for iterating it parallely

https://www.mkyong.com/java8/java-8-collectors-groupingby-and-mapping-example/
https://dzone.com/articles/the-ultimate-guide-to-the-java-stream-api-grouping
Function identity - http://www.java2s.com/Tutorials/Java/java.util.function/Function/1080__Function.identity.htm
https://www.javabrahman.com/java-8/java-8-java-util-function-function-tutorial-with-examples/
Collectors.counting - https://www.geeksforgeeks.org/java-8-collectors-counting-with-examples/

Default methods in interfaces to define the body of the method with **default** keyword in function signature
Public Interface class{
    Default method() {}
    }

Functional interface - interface contain single abstract method

Before java 8, if u take classes like runnable, comparable have many methods and 1 abstract method like run, comparable to implement in subclasses, now we can use lambda expression
    Without writing anonymous inner classes or
    Implementation of interfaces

Lambda expression

Method reference - Collection::stream, System.out::println

Optional.of
Optiona.empty()
Optional.ofNullable(value)

flatMap - flattening the list of list elements

Supplier - get()
Consumer - accept() - equal to set

UnaryOperator T func(T)
Binaryoperator T func(T,T)
Function Y function(X)
BiFunction Y function(X,Z)
Predicate - equals to assert() method which returns boolean by taking some parameters
Boolean func(X, Y)

Java nio buffers - non blocking input/output buffers
https://www.javatpoint.com/java-nio-buffers

Creating own annotation in java
https://www.mindstick.com/articles/12141/annotations-in-java-target-and-retention

Ref:
flatMapcheck, SampleTest, flatMapCheck1.java
FunctionalInterfaces.java
MethodReference.java
CntDuplicatesUsingStream.java

Docker advantages

23 July 2019 17:28

Lightweight than virtual machines
Employed for managing containers
Containers Detached from each other
Bundle their own tool, softwares, libraries and configuration files
Scalablity
Speed - installation of docker & pulling the images
Modularity - Application become modular since there is the isolation for each components

Usage of RUN, CMD & ENTRYPOINT

2 forms
- Shell form
- Executable - exec form ['bin/bash', cmd1, arg1, arg2]

CMD - default command execution
- Will be ignored if you are passing any explicit commands and arguments

ENTRYPOINT - always commands will be executed
- We can pass additional arguments using CMD by CMD[arg1, arg2]
- Also we can pass arguments while running docker run command also

Questions

31 July 2019 12:36

Multithreading

Synchronized static and non-static methods

Lock state will be different for Synchronized static and non-static methods

Class level lock and Object level lock

Hash Collisions - 2 object hashing return same address references, it will be resorted by creating linkedlist

Time Complexity

03 August 2019 18:49

Sorting algorithms

algorithms	Time complexity
For(int i=o;i<n;i++) { }	O(n) - 1 loop(n)

For(int i=0;i<n;i++) {
For(int j=0;i<n;j++) {
}
}
	o(n*n) - 2 loops
	
	Selection sort, bubble sort,
	insertion sort

For(int i=0;i<n;i++) {
}
For(int j=0;j<m;j++) {
}
	O(n) + O(m) = O(n+m)
	
	If n==m, O(2n)

While(low <= high) {
Mid = (low+height)/2
If(target < list[mid]/2){
High = mid-1;
} if(target > list[mid]/2){
Low = mid+1;
Else break;
	Binary search, Binary Tree
	
	O(log(N))
	N - no of elements in list
	
	Divide & Conquier

For(int i=0;i<n;i++){
Divide&conquer
}
	Combination of loop &
	divide and conquer
	And choosing the pivot everytime
	
	Log(N) - if divided/multiplied by common amount
	
	O(N * log(N))
	
	Quick sort & Merge Sort, Heap Sort

For(int i=0; i<N; pow(I,c)) {
}
Func©
	LogLog(N) - if the loop exponentially increased/reduced
	By const amount N
	
	Pow(I,c), sqrt, cuberoot of N,

https://www.studytonight.com/data-structures/time-complexity-of-algorithms
https://www.geeksforgeeks.org/time-complexities-of-all-sorting-algorithms/

QuickSort -- preferred for Arrays,
MergeSort - linked lists

QuickSort - in-sort algorithm,
MergeSort requires extra space
Allocating & deallocating in extra space increases time complexity
Link : https://www.geeksforgeeks.org/why-quick-sort-preferred-for-arrays-and-merge-sort-for-linked-lists/

Datastructures

09 September 2019 12:40

https://www.geeksforgeeks.org/given-an-array-arr-find-the-maximum-j-i-such-that-arrj-arri/

- traverse 2 loops - complexity - O(n) + O(n) - 2loops
- one array to arrange data from low to high
- another array to arrange data from high to low
- now using merge sort
  - check if arr[i]<arr[j]
    - then maxDiff = max(maxDiff, j-i)
      j = j+1
  - else
    - i = i+1

https://www.geeksforgeeks.org/detect-and-remove-loop-in-a-linked-list/

Find the point where the slow pointer and fast pointer met
Calculate the number of nodes in a loop
Now to find the point where the loop starts,
move one pointer - Pointer1 from the index 0 until the count of loop
Now move another pointer - pointer2 from index 0 and move the pointer1 at the same time with the same steps,
The point where the 2 pointers meet is the starting point and one point before that pointer is the nodes which need to be delinked to break the loop

https://www.geeksforgeeks.org/design-a-stack-that-supports-getmin-in-o1-time-and-o1-extra-space/

Push(x) -
T = 2*x - minEle
Stack.push(t)
Pop() -
T = stack.pop()
Val = 2*minEle - t

https://www.careercup.com/question?id=5728764059713536
https://www.gohired.in/2015/10/03/walmart-labs-interview-questions-experience/
https://www.geeksforgeeks.org/search-in-row-wise-and-column-wise-sorted-matrix/
https://www.geeksforgeeks.org/clone-an-undirected-graph/
https://www.geeksforgeeks.org/kth-smallestlargest-element-unsorted-array/ - O(n) - performance using quicksort to find the sorted data
QuickSort - https://www.youtube.com/watch?v=7h1s2SojIRw -
In memory sorting
Selecting the pivot - mostly at the end of array,
Comparing the each data with the pivot and arrange the cells in such a way that the elements before pivot are lesser and elements after pivot are larger
and fit the pivot properly at the time of index which is placed permanently
Divide the array by keeping pivot at the boundary one so do the in-place sorting consecutively
At the end, all the elements in the array will be sorted

MergeSort - dividing it completely into individual cells and merge them by sorting them at the time of merging it.
It doesn't do in-memory sorting, so additional memory is required to do it.

https://www.geeksforgeeks.org/print-nodes-distance-k-given-node-binary-tree/ -
Find the target from the root node by searching in left and right subtree
Once u find the target, print the downstream data from the target node

Also go the call back of stack one by one,

Either go straight in which ever direction first or

go to the opposite side of the target node and try to go the extent of leaf nodes until it reaches the particular distance

https://www.geeksforgeeks.org/reverse-level-order-traversal/ -

using queue & stack - by storing queue from right to left order, and store in stack

By measuring height of the tree

Height of the tree can be measured by recursing from left and right and take the max height from left to right

Recursively print once the height reaches 1 from top to bottom

https://www.geeksforgeeks.org/level-order-tree-traversal/

using queue

By measuring height of the tree

Height of the tree can be measured by recursing from left and right and take the max height from left to right

Recursively print it from 1 to max height of tree

https://www.geeksforgeeks.org/level-order-traversal-in-spiral-form/

using 2 stacks. Insert the data alternatively in 2 stacks

By measuring height of the tree

Height of the tree can be measured by recursing from left and right and take the max height from left to right

Recursively print it from 1 to max height of tree and using Boolean to print left to right or right to left alternatively

https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/

https://www.geeksforgeeks.org/search-in-a-row-wise-and-column-wise-sorted-2d-array-using-divide-and-conquer-algorithm/

https://stackoverflow.com/questions/52787803/given-a-node-how-long-will-it-take-to-burn-the-whole-binary-tree

Spring security

19 September 2019 23:12

https://spring.io/guides/gs/securing-web/

https://www.baeldung.com/spring-boot-security-autoconfiguration

https://www.baeldung.com/spring-security-expressions

https://www.baeldung.com/spring-security-authentication-provider

https://www.javainuse.com/spring/sprsec
https://www.javainuse.com/spring/boot_form_authentication_handler
https://www.javainuse.com/spring/boot_security_jdbc_authentication
https://www.javainuse.com/spring/boot-jwt-mysql
https://www.javainuse.com/spring/boot-jwt

Spring security method level
http://www.studytrails.com/frameworks/spring/spring-security-method-level/
https://howtodoinjava.com/spring-security/spring-3-method-level-security-example-using-preauthorize-and-secured/

```
@EnableGlobalMethodSecurity(securedEnabled = true)
```

Java pcf

22 September 2019 22:51

https://www.javainuse.com/pcf/pcf-hello