EXPERIMENT -1

ASHISH PATIL
D11AD-46

AIM:- To understand DevOps: Principles, Practices, and DevOps Engineer Role and Responsibilities

Theory:-

DevOps, a portmanteau of "development" and "operations," is a set of practices that aims to automate and improve the collaboration between software development and IT operations teams. The primary goal of DevOps is to shorten the development lifecycle and deliver high-quality software continuously. Here's a detailed overview of key aspects of DevOps:

1. Definition and Philosophy:
- Collaboration: DevOps emphasizes collaboration and communication between development (Dev) and operations (Ops) teams. This breaks down silos and promotes a more integrated and efficient workflow.
- Automation: Automation is a cornerstone of DevOps. It involves the use of tools and scripts to automate manual and repetitive tasks, reducing errors and improving efficiency.

2. Key Principles:
- Continuous Integration (CI): Developers integrate code into a shared repository multiple times a day. Each integration triggers automated tests to ensure that the new code doesn't break existing functionality.
- Continuous Delivery (CD): The code is always in a deployable state, and any changes can be released to production at any time. CD extends CI by automatically deploying code to testing, staging, and production environments.

3. DevOps Lifecycle:
- Planning: Define the scope, objectives, and requirements of the software development process.
- Development: Write, test, and integrate code collaboratively.
- Testing: Automate testing to identify and fix issues early in the development process.
- Deployment: Release code to different environments using automation.
- Monitoring and Feedback: Continuously monitor the application in production, gather feedback, and make improvements.

4. Key Practices and Tools:
- Version Control: Tools like Git enable developers to manage and track changes in the source code.
- Continuous Integration Tools: Jenkins, Travis CI, and GitLab CI are examples of tools that automate the process of integrating code changes.
- Configuration Management: Tools like Ansible, Chef, and Puppet automate the configuration and management of infrastructure.

- Containerization and Orchestration: Docker is widely used for containerization, and Kubernetes is a popular orchestration tool to manage and scale containerized applications.
- Monitoring and Logging: Tools like Prometheus, ELK Stack (Elasticsearch, Logstash, Kibana), and Grafana help monitor and log system behavior.

5. Cultural Aspects:

- Collaboration: Encourages open communication and collaboration between development, operations, and other stakeholders.
- Shared Responsibility: Teams share responsibility for the entire software development lifecycle, from planning to deployment.

6. Benefits of DevOps:

- Faster Delivery: Accelerates the software development and delivery process.
- Improved Collaboration: Breaks down traditional silos between development and operations teams.
- Increased Efficiency: Automation reduces manual errors and speeds up repetitive tasks.
- Greater Stability: Continuous testing and monitoring help catch and fix issues early.

7. Challenges:

- Cultural Resistance: Changing the way teams work can face resistance.
- Tool Complexity: The plethora of tools available can be overwhelming.
- Security Concerns: Rapid development and deployment can sometimes compromise security if not properly managed.

DevOps is a holistic approach to software development and delivery, fostering collaboration, automation, and continuous improvement to achieve more reliable and efficient processes. It is essential for organizations looking to stay competitive in the fast-paced world of software development.

DEVTOOLS:-
Tool used:- Redis.

Redis is an open-source, in-memory data structure store that can be used as a database, cache, and message broker. It is known for its speed, simplicity, and versatility. Here's a comprehensive overview of Redis:

1. Key Features:

- In-Memory Data Store: Redis primarily stores data in RAM, which allows for extremely fast read and write operations.
- Data Structures: Supports various data structures such as strings, hashes, lists, sets, and sorted sets, providing flexibility for different use cases.
- Persistence: While it is an in-memory store, Redis can be configured to persist data to disk, ensuring durability.
- Atomic Operations: Redis commands are atomic, meaning they either succeed entirely or have no effect.
- Replication: Supports master-slave replication for high availability and fault tolerance.

2. Data Structures in Redis:

- Strings: Simple key-value pairs.
- Hashes: Key-value pairs within a key, useful for representing objects.
- Lists: Collections of ordered elements (linked lists).
- Sets: Unordered collections of unique elements.
- Sorted Sets: Similar to sets but with an associated score, allowing for range queries.

3. Use Cases:

- Caching: Redis is often used as a cache due to its fast read and write operations.
- Session Store: Storing session data in-memory for quick access.
- Real-time Analytics: Due to its speed, Redis is used in scenarios where low-latency data processing is crucial.
- Pub/Sub Messaging: Redis supports publish/subscribe for building real-time messaging systems.

4. Persistence:

- RDB Snapshots: Periodically saves a snapshot of the data to disk.
- AOF (Append-Only File): Logs every write operation to a file, allowing for data recovery in case of a crash.
- Combination: Users can configure both RDB snapshots and AOF for different levels of persistence.

5. Replication:

- Master-Slave Replication: Allows for data redundancy and high availability. The master serves write operations, and one or more slaves replicate the data.
- Automatic Failover: Redis Sentinel can be used for automatic master promotion if the master node fails.

6. Partitioning:

- Sharding: Redis can be horizontally scaled by partitioning data across multiple nodes.
- Consistent Hashing: Ensures that keys map consistently to the same shard, minimizing data movement during scale-up or scale-down.

7. Transactions:

- Atomic Transactions: Multiple commands can be executed as a single transaction, ensuring atomicity.
- Isolation: Transactions in Redis are atomic, but they are not isolated (no rollback if one operation in a transaction fails).

8. Security:

- Authentication: Supports password-based authentication.
- Network Security: Users can configure Redis to bind only to specific network interfaces.

9. Community and Ecosystem:

- Active Community: Redis has a large and active community that contributes to its development.
- Client Libraries: Provides client libraries for various programming languages, making it easy to integrate with different applications.

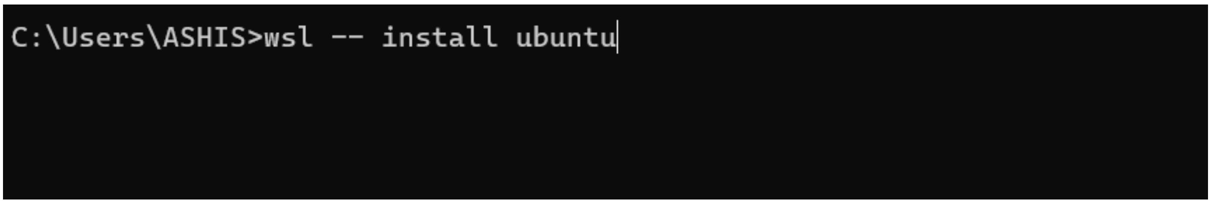
10. Limitations:

- Size Limitations: Limited by the amount of available RAM.
- Complex Queries: Not as suitable for complex queries compared to traditional relational databases.

Redis is a powerful and versatile tool used by many organizations for caching, real-time analytics, and other scenarios where low-latency access to data is critical. Its simplicity and speed make it a popular choice for various use cases.

Requirements:-
Linux,macOS

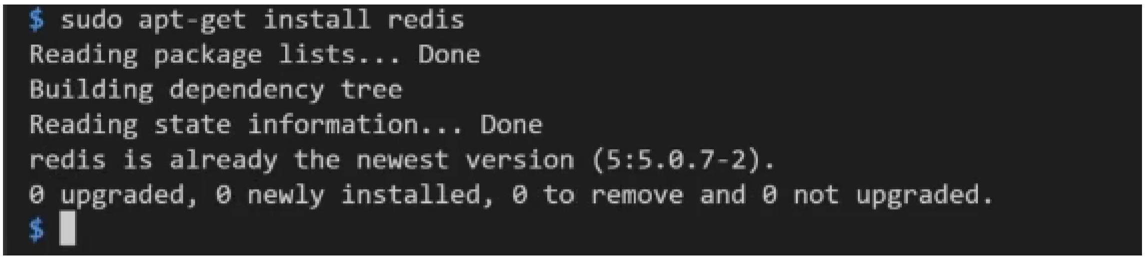
STEP-1:
Installing wsl on windows using



```
C:\Users\ASHIS>wsl -- install ubuntu
```

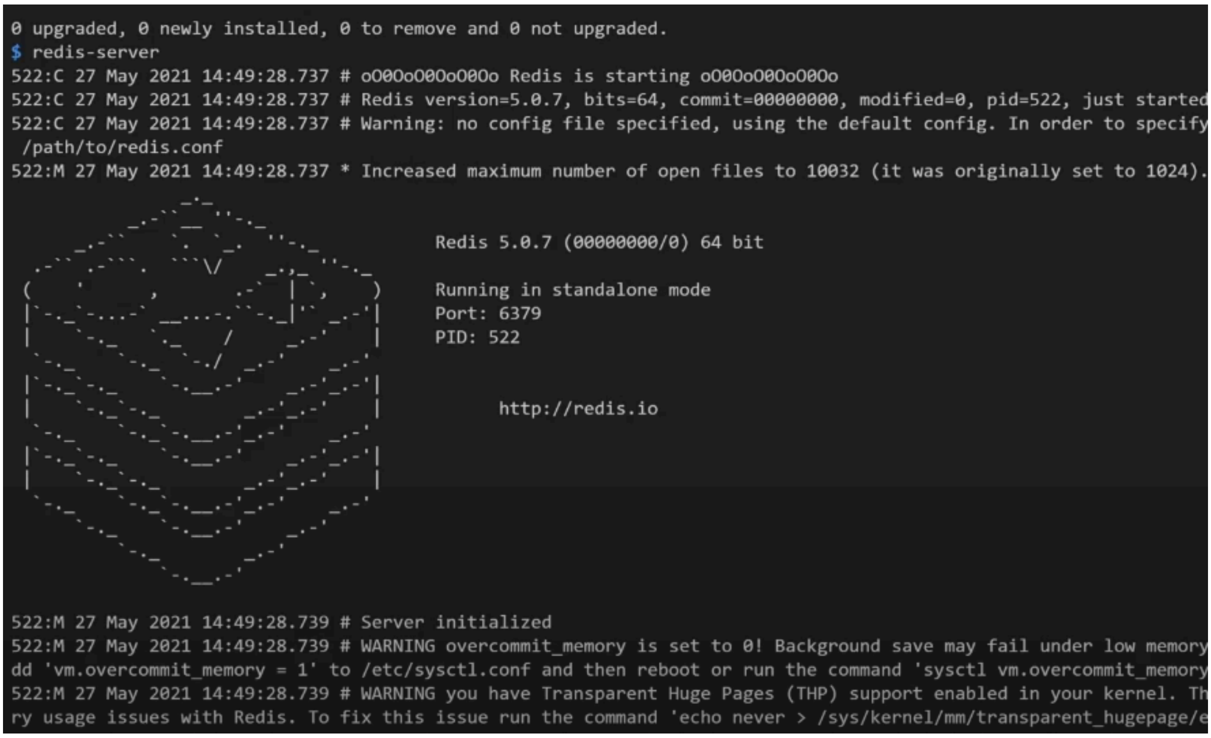
Installing redis using

Sudo apt install redis-server



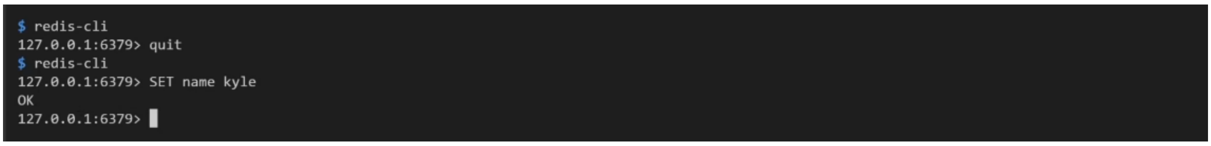
```
$ sudo apt-get install redis
Reading package lists... Done
Building dependency tree
Reading state information... Done
redis is already the newest version (5:5.0.7-2).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
$ 
```

Starting redis



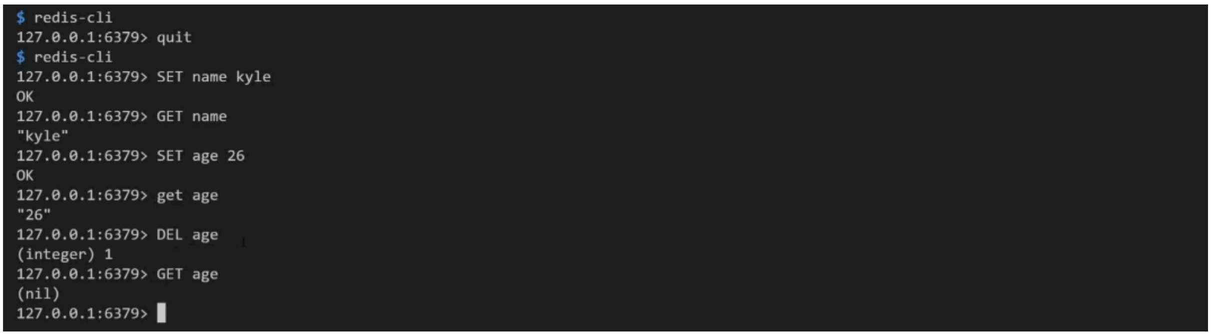
```
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
$ redis-server
522:C 27 May 2021 14:49:28.737 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
522:C 27 May 2021 14:49:28.737 # Redis version=5.0.7, bits=64, commit=00000000, modified=0, pid=522, just started
522:C 27 May 2021 14:49:28.737 # Warning: no config file specified, using the default config. In order to specify
 /path/to/redis.conf
522:M 27 May 2021 14:49:28.737 * Increased maximum number of open files to 10032 (it was originally set to 1024).
                _._
           _.-``__ ''-._
      _.-``    `.  `_.  ''-._           Redis 5.0.7 (00000000/0) 64 bit
  .-`` .-```.  ```\/    _.,_ ''-._
 (    '      ,       .-`  | `,    )     Running in standalone mode
 |`-._`-...-` __...-.``-._|'` _.-'|     Port: 6379
 |    `-._   `._    /     _.-'    |     PID: 522
  `-._    `-._  `-./  _.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |           http://redis.io
  `-._    `-._`-.__.-'_.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |
  `-._    `-._`-.__.-'_.-'    _.-'
      `-._    `-.__.-'    _.-'
          `-._        _.-'
              `-.__.-'

522:M 27 May 2021 14:49:28.739 # Server initialized
522:M 27 May 2021 14:49:28.739 # WARNING overcommit_memory is set to 0! Background save may fail under low memory
dd 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory
522:M 27 May 2021 14:49:28.739 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. Th
ry usage issues with Redis. To fix this issue run the command 'echo never > /sys/kernel/mm/transparent_hugepage/e
```

Starting redis cli:

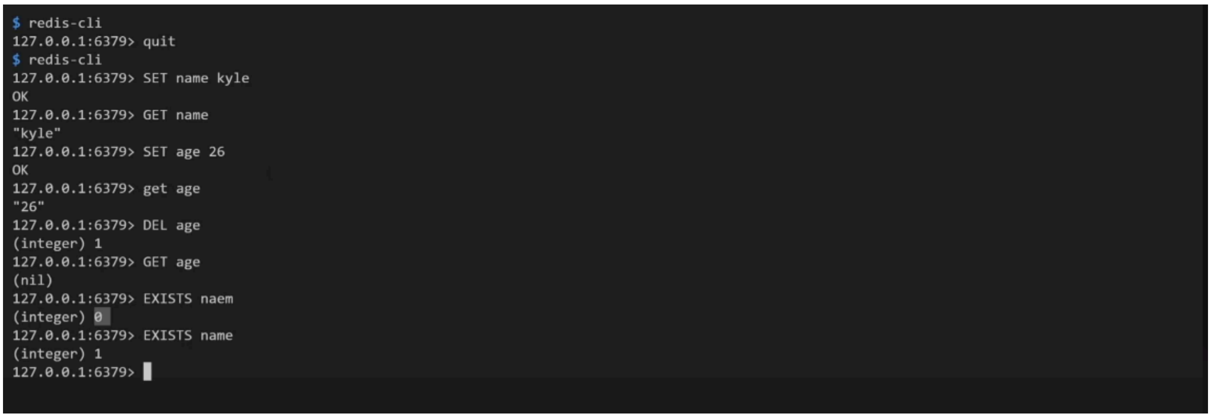


```
$ redis-cli
127.0.0.1:6379> quit
$ redis-cli
127.0.0.1:6379> SET name kyle
OK
127.0.0.1:6379> 
```

Basic Redis Operations:-

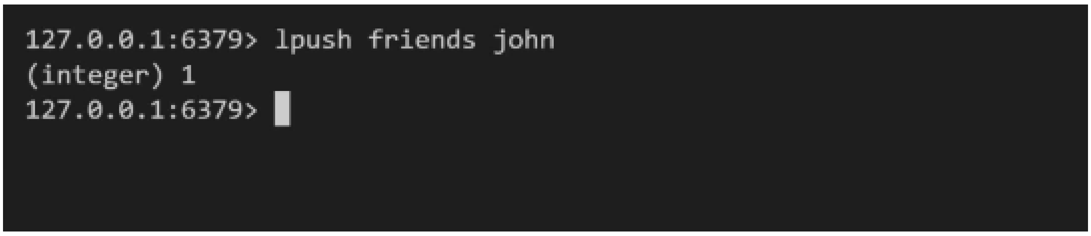


```
$ redis-cli
127.0.0.1:6379> quit
$ redis-cli
127.0.0.1:6379> SET name kyle
OK
127.0.0.1:6379> GET name
"kyle"
127.0.0.1:6379> SET age 26
OK
127.0.0.1:6379> get age
"26"
127.0.0.1:6379> DEL age
(integer) 1
127.0.0.1:6379> GET age
(nil)
127.0.0.1:6379> 
```



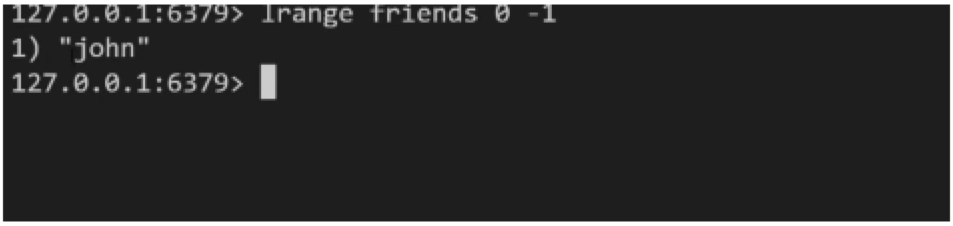
```
$ redis-cli
127.0.0.1:6379> quit
$ redis-cli
127.0.0.1:6379> SET name kyle
OK
127.0.0.1:6379> GET name
"kyle"
127.0.0.1:6379> SET age 26
OK
127.0.0.1:6379> get age
"26"
127.0.0.1:6379> DEL age
(integer) 1
127.0.0.1:6379> GET age
(nil)
127.0.0.1:6379> EXISTS naem
(integer) 0
127.0.0.1:6379> EXISTS name
(integer) 1
127.0.0.1:6379>
```

Using lists:



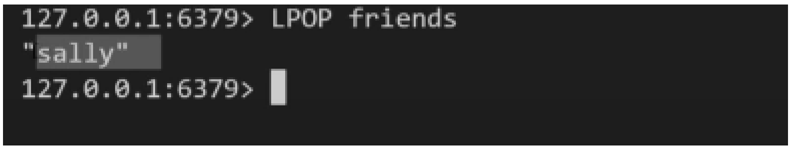
```
127.0.0.1:6379> lpush friends john
(integer) 1
127.0.0.1:6379>
```

Viewing Lists:



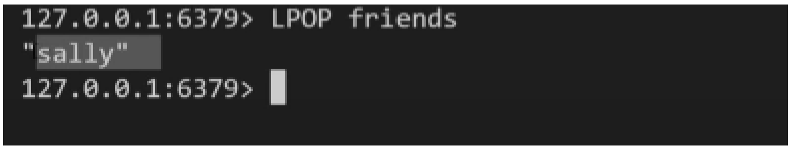
```
127.0.0.1:6379> lrange friends 0 -1
1) "john"
127.0.0.1:6379>
```

Popping from above in a list:

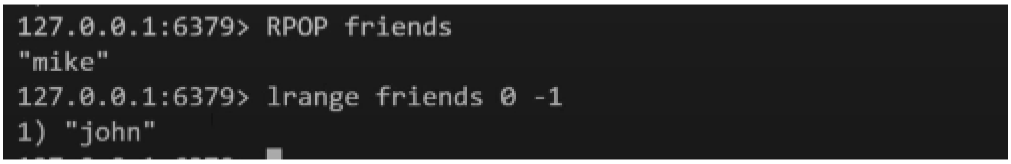


```
127.0.0.1:6379> LPOP friends
"sally"
127.0.0.1:6379>
```

Popping from back:

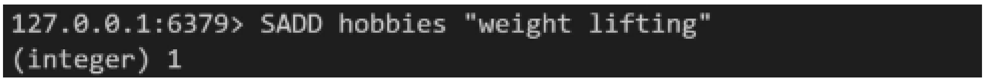


```
127.0.0.1:6379> LPOP friends
"sally"
127.0.0.1:6379>
```

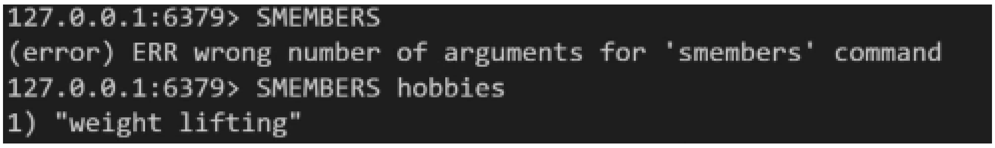


```
127.0.0.1:6379> RPOP friends
"mike"
127.0.0.1:6379> lrange friends 0 -1
1) "john"
```

Working with Sets:



```
127.0.0.1:6379> SADD hobbies "weight lifting"
(integer) 1
```



```
127.0.0.1:6379> SMEMBERS
(error) ERR wrong number of arguments for 'smembers' command
127.0.0.1:6379> SMEMBERS hobbies
1) "weight lifting"
```

Conclusion: We have successfully installed redis and discovered DevOps as a concept and also learned and used a DevOps tool.