IT认证电子书

质　量　更　高　　服　务　更　好

半年免费升级服务

http://www.itrenzheng.com

Exam : **EX210**

Title : Red Hat Certified Specialist in Cloud Infrastructure

Version : DEMO

1.You are responsible for managing the OpenStack control plane and ensuring all core services are running without failure. A user reports that OpenStack Compute (Nova) is not responding. Your task is to check the status of all OpenStack services on the control plane, restart any failed services, and verify that they are running correctly.
How would you achieve this?
Answer:
1. Log in to the OpenStack Controller node as root: ssh root@controller-node
2. Check the status of all OpenStack-related services:
systemctl list-units --type=service --state=running | grep openstack
3. Identify any failed services by running: systemctl list-units --type=service --state=failed
4. If the nova-api or any other service is down, restart it: systemctl restart openstack-nova-api
5. Verify the service has started successfully and check logs for issues:
systemctl status openstack-nova-api
journalctl -xe -u openstack-nova-api
Explanation:
Ensuring that OpenStack services are running is critical for the stability of the cloud infrastructure. The systemctl list-units command helps identify running and failed services, and restarting them ensures minimal downtime. The journalctl -xe command provides detailed logs to debug issues related to service failures. Regular monitoring of service statuses helps in proactive issue resolution.

2.As part of routine maintenance, you need to check and confirm whether the OpenStack Identity (Keystone) service is functional and responding correctly.
How would you verify the Keystone service's status, check if it is accessible via API, and restart it if necessary?
Answer:
1. Check Keystone service status on the controller node: systemctl status openstack-keystone
2. If the service is not running, restart it:
systemctl restart openstack-keystone
3. Verify API connectivity by running a token authentication request: openstack token issue
4. If the API is not responding, check the logs for errors:
journalctl -xe -u openstack-keystone
5. Ensure the Keystone endpoints are properly configured: openstack endpoint list | grep identity
Explanation:
Keystone is the authentication and authorization service for OpenStack, and its failure can impact all services. The openstack token issue command ensures Keystone is operational. Checking logs helps identify common issues such as database connectivity problems or expired tokens. Restarting the service can resolve temporary failures and restore authentication functionality.

3.Your OpenStack cloud is experiencing intermittent failures, and you suspect issues with the RabbitMQ messaging service, which is critical for communication between OpenStack components.
How would you check the RabbitMQ service status, troubleshoot issues, and restart it if necessary?
Answer:
1. Check the status of RabbitMQ service on the controller node: systemctl status rabbitmq-server
2. If the service is down, restart it:

systemctl restart rabbitmq-server
3. Verify RabbitMQ cluster status to ensure nodes are communicating:
rabbitmqctl cluster_status
4. If queues are blocked or unresponsive, list all queues: rabbitmqctl list_queues
5. Check logs for potential errors such as connection failures or resource limits: journalctl -xe -u rabbitmq-server

Explanation:

RabbitMQ is a core messaging component for OpenStack services like Nova, Neutron, and Cinder. If it fails, inter-service communication breaks down, leading to cloud instability. Monitoring cluster status and checking queues help diagnose issues like message backlogs. Restarting RabbitMQ can resolve temporary failures, but logs should be analyzed for persistent problems.

4.You need to back up the OpenStack database before applying a system update.
How would you create a full backup of the OpenStack MariaDB database while ensuring minimal downtime?

Answer:

1. Log in to the database node or controller node where MariaDB is running: ssh root@controller-node
2. Check the current database size and ensure sufficient space for backup: du -sh /var/lib/mysql
3. Stop all OpenStack services to prevent inconsistent backup:
systemctl stop 'openstack-*'
4. Perform a database dump using mysqldump:
mysqldump -u root -p --all-databases > /var/backups/openstack_backup.sql
5. Restart OpenStack services after backup completion:
systemctl start 'openstack-*'
6. Verify the backup file integrity:
ls -lh /var/backups/openstack_backup.sql

Explanation:

Backing up the OpenStack database is essential before performing upgrades or maintenance. Using mysqldump ensures all databases are backed up, while stopping OpenStack services prevents inconsistencies. Restarting services after backup maintains availability. Regular backups protect against data loss and corruption during upgrades.

5.You need to restore the OpenStack database from a previously created backup due to data corruption.
How would you restore the MariaDB database while ensuring the OpenStack services can access it correctly?

Answer:

1. Log in to the database node or controller node: ssh root@controller-node
2. Stop all OpenStack services to prevent issues during restoration: systemctl stop 'openstack-*'
3. Drop the current database to clear corrupted data:
mysql -u root -p -e "DROP DATABASE openstack;"
4. Restore the database from the backup file:
mysql -u root -p < /var/backups/openstack_backup.sql
5. Start OpenStack services and validate functionality:
systemctl start 'openstack-*'

openstack service list
6. Check the database logs for errors: journalctl -xe -u mariadb

Explanation:

Restoring a database from backup is critical when data corruption occurs. Stopping services prevents conflicts during restoration. Dropping and reloading the database ensures a clean import. Checking logs post-restoration helps identify residual issues. Always validate OpenStack functionality after restoring data.