Project:
Plan, Reduce, Repeat

Directions and Submission Template

César Blanco Fernández:
16.02.2023

Overview:

You have recently joined the SRE team for an exotic plant reseller startup. They already have a small SRE team in place consisting of two other members. You are just finishing up your training period and are now ready to be on your own.

You have a busy week ahead of you as there is a release this week plus your on-call shift. Part of your release duties includes helping to maintain the as-built document by adding this new release, as well as planning for system resource changes. For your on-call shift, you have to respond to alerts as they come in and write up an on-call summary to document your shift. Finally, you'll round out your week by helping to reduce toil. You will have to identify any toil you encounter throughout the week and create a toil reduction plan. After you have a plan all ready, you will need to work on implementing that plan by writing some scripts to help automate tasks.

Scenario 1

Release Day

Release Night

Summary

Tonight is release night, and it will be your first time assisting with a release as an SRE. The process now is manual, with no real consideration for how releases may impact resource allocation. Luckily, your other team members have started implementing an as-built document. You'll have to add tonight's release to the document. The release is a pretty major release with the addition of a new feature that will bring in a large number of new clients. Looking at the results from testing, you can see that this new feature is going to add additional resource requirements as it is both more memory and, to a lesser extent, CPU intensive than before.

Current Release Features

This release will have the following changes that will need to be documented on the as-built design document. The developers have been hard at work implementing the following tickets:

- Ticket 203 added a new catalog for exotic plants. This ticket added new tables in the database to handle the additional catalogs.
- Ticket 202 rearranged the catalog menu in the UI to accommodate the additional catalog, as well as making it more user-friendly.
- Ticket 201 added an additional component to the application, an order processor. The order processor is responsible for batch processing orders on a schedule. The reasoning behind this was to decouple the UI from order processing, and since order processing can be CPU intensive, this decoupling prevents the app from performing poorly. The Design Doc 5247 goes into more detail about the design specifics.
- Ticket 205 fixed a security flaw where attackers could execute a SQL injection attack.

Release Night, cont.

Release Process

The established release process is a manual affair generally done by one of the operations team members. The OPs team generally will download the latest code, shut down the app, run the database migrations, change or add any needed configurations and then start the app back up. In the past this has caused issues as steps have been forgotten, not all the scripts were executed, the app was not restarted properly, among other issues. During the release window, the OPs engineer would also add new resources as needed. This has led to downtime in the past as the app became overloaded and could not serve requests anymore.

Release Planning

During load testing for this release, it was determined that

- Main Application
    - The new catalog feature increases RAM usages by 25% for the same number of users, while not increasing CPU significantly. Currently, the main application containers are utilizing almost 85% of the RAM allocated.
    - At the current resource allocation, each server can handle 500 concurrent users. Currently, there are 3 application containers to support about 2000 total users, with about 1300 being on at any one time. This release is expected to add about 1.5 to 2.5 times the total number of users, with a need to handle 2600 users concurrently.
- Order Processor
    - This component has a high CPU utilization with moderate RAM requirements. In testing, a full loaded queue used approximately 1 Gb of RAM.
    - The component runs with 2 concurrent processes, pulling orders out of the database and processing them for fulfillment. This component can process 4 orders at a time, with the average order taking between 10 and 15 seconds to complete depending on the size and complexity of the order as well as CPU resource allocation. QA recommends twice the CPU as the main application.
- Database
    - The database was provisioned to handle a much larger application than what the company has now and passed the load tests with flying colors.

As-Built Doc Template
Release Version

Stakeholders

These are the teams and members involved in this reason. This should include ops members, developers, SRE members, database admin, etc

Code Changes

This section should include a list of code changes going into this release separated into groups (for example, by bug fix, feature addition, and security fixes). This should be a short summary of the change with a ticket included to follow up with for more detailed information.

Data and System Changes

This should be formatted similarly to the code changes section, except listing any changes to the data model (database or API changes) or system changes.

Design decision highlights

Document the high-level reasoning behind any design choices. This section should only include a summary of the design decision with links to supporting documentation to follow up with for more detailed information.

Test Section

In this section, list any notable highlights from testing. Things to include here would be any changes to the testing methodology, changes to the test performed, and any tests that are not currently pass (or pass with a warning).

Deployment Notes

Include any changes made to the deployment process or any changes that should be made to improve in feature releases.

As-Built Doc
Release 1

Stakeholders

- Developers
    - John Doe
    - ane Peters
    - Sam Ross
- Ops
    - Jay Smith
- SRE
    - John Robert

Code Changes

- Security fixes
    - Added new password requirements (Tk-100)
    - Fixed how SQL queries were handled (Tk-103)
- Feature Additions
    - Added new menu options for users (Tk-102)
    - Users can now have middle names (Tk-101)

Data and System Changes

- Data model changes
    - Added columns for middle names in user table (TK-101)
    - Added additional New Menu table (Tk-102)
    - Users table was split into 2 smaller tables (TK-101)

As-Built Doc
Release 1

Design decision highlights

Users table was split into two smaller tables to create more efficient queries and mappings. Keeping it as one big table began to cause slow queries and allowed for a larger number of users. See Design Doc 134 for further discussion.

Test Section

All test suites are passing 100%.

Deployment Notes

The database admins asked for an additional set of scripts to be run for data corrections.

Deployment File
Release 1

```
ApiVersion: apps/v1
kind: Deployment
metadata:
  name: app-deployment
  namespace: course4
  labels:
    app: mainApp
spec:
  replicas: 3
  selector:
    matchLabels:
      app: mainApp
  template:
    metadata:
      labels:
        app: mainApp
    spec:
      containers:
      - name: mainApp
        image: nginx:latest
        resources:
          requests:
            memory: 256mb
            cpu: 250m
        ports:
        - containerPort: 80
```

As-Built Doc
Release 2

Continue the As-Built Doc for this new release here. Add more pages if needed.

Stakeholders

- Developers
    - John Doe
    - ane Peters
    - Sam Ross
- Ops
    - Jay Smith
- SRE
    - John Robert

Code Changes

- Security fixes
    - Fixed a security flaw where attackers could execute a SQL injection attack (Tk 205)
- Feature Modification
    - Rearranged the catalog menu in the UI to accommodate the additional catalog (Tk-202)

Data and System Changes

- Data model changes
    - Added a new catalog for exotic plants (TK-203)
- System changes
    - Added an additional component to the application, an order processor (Tk 201)

As-Built Doc
Release 2

Design decision highlights

Added an order processor to the application. The order processor is responsible for batch processing orders on a schedule. The reasoning behind this was tho decouple the UI from order processing, and since order processing can be CPU intensive, this decoupling prevents the app from performing poorly. The Design Doc 5247 for further discussion.

Test Section

All test suites are passing 100%.

Deployment Notes

1. Main Application

    A. RAM: 85% of the RAM allocated + increase RAM usages by 25% for the same number of users. It is necessary 512 MB RAM size

    - (0.85 x 256mb) + (0.25 x 256mb) = 281,6mb is necessary. The result isn't a valid RAM size, the nearest RAM size is 512 mb

    B. CPU: not increasing significantly

    C. It is necessary to increase the replicas number to 6

    - Because: 6 replicas x 500 users = 3000 users > 2600 concurrents users

As-Built Doc
Release 2

Deployment Notes (continue)

2. Order Processor

   A. RAM: 2048 MB. It was selected even if the load testing should that the order processor used approximately 1024mb, this is because:

      1. We don’t want 100% RAM utilization
      2. With appropriate monitoring in place, we might decide to scale the RAM down ro to keep it the same

   B. CPU: QA recommend twice the CPU as the main application (500 MB)

3. Database

   - Requires no changes.

Deployment File
Release 2

```
ApiVersion: apps/v1
kind: Deployment
metadata:
  name: app-deployment
  namespace: course4
  labels:
    app: mainApp
spec:
  Replicas: 6
  selector:
    matchLabels:
      app: mainApp
  template:
    metadata:
      labels:
        app: mainApp
    spec:
      containers:
      - name: mainApp
        image: nginx:latest
        resources:
          requests:
            memory: 512mb
            cpu: 250m
        ports:
        - containerPort: 80
      - name: order_processor
        image: nginx:latest
        resources:
          requests:
            memory: 2048mb
            cpu: 500m
        ports:
        - containerPort: 80
```

Scenario 2

On-Call Shift

On-Call Shift

Summary

Today is your first on-call shift as an SRE. During your shift, you will have to respond to alerts to keep the system running at its best using the on-call best practices learned in this course. During your on-call shift, make sure to be thinking of ways to reduce toil. After your on-call shift is over, you will be responsible for writing a summary of your shift and a post-mortem. On the following slides you will encounter several different “alerts” from your monitoring stack. Each “alert” will contain several different parts that will help you write your on-call log for your shift. Additionally, you’ll encounter an application outage that will require a post-mortem.

Alert Components

Summary -- This will be general knowledge about the systems involved that you would know if you had actually been working at the company. It will include a brief description of the systems involved as well information about how it is managed.

Standard Operating Procedure (SOP) -- This will be a short description of the steps to troubleshoot and potentially correct the cause of the alert.

Log and Monitoring Details -- This section will contain snippets of relevant logs and monitoring data (graphs, metrics, etc.) that are associated with responding to an alert.

On-call Log

After your on-call shift you’ll need to add to the on-call log. There is a provided sample template for you to use that includes all the necessary fields. Remember your on-call log is used to help track recurring alerts/issues as well as providing a record of the steps taken to resolve the issue.

Post-Mortem

Unfortunately there will be an application outage on your shift that will require a post-mortem. You will only be responsible for filling in your involvement, plus you’ll be in charge of creating an action plan and impact assessment.

On-Call Shift -- Alert 1

Order Processing Issues

Summary

You receive an alert that the number of Outstanding Orders is too high. Orders are processed by a separate component from the main application. It runs periodically (every hour currently) to batch process any open orders. Your team has set up some monitors to keep track of how well the order processor is doing.

SOP

Number of Outstanding Orders is Too High

If this alert comes through you will need to check the dashboard to see if the Order Processor is overloaded with orders. If there is a high number of orders contact Ops to see if the processor should be run more frequently.

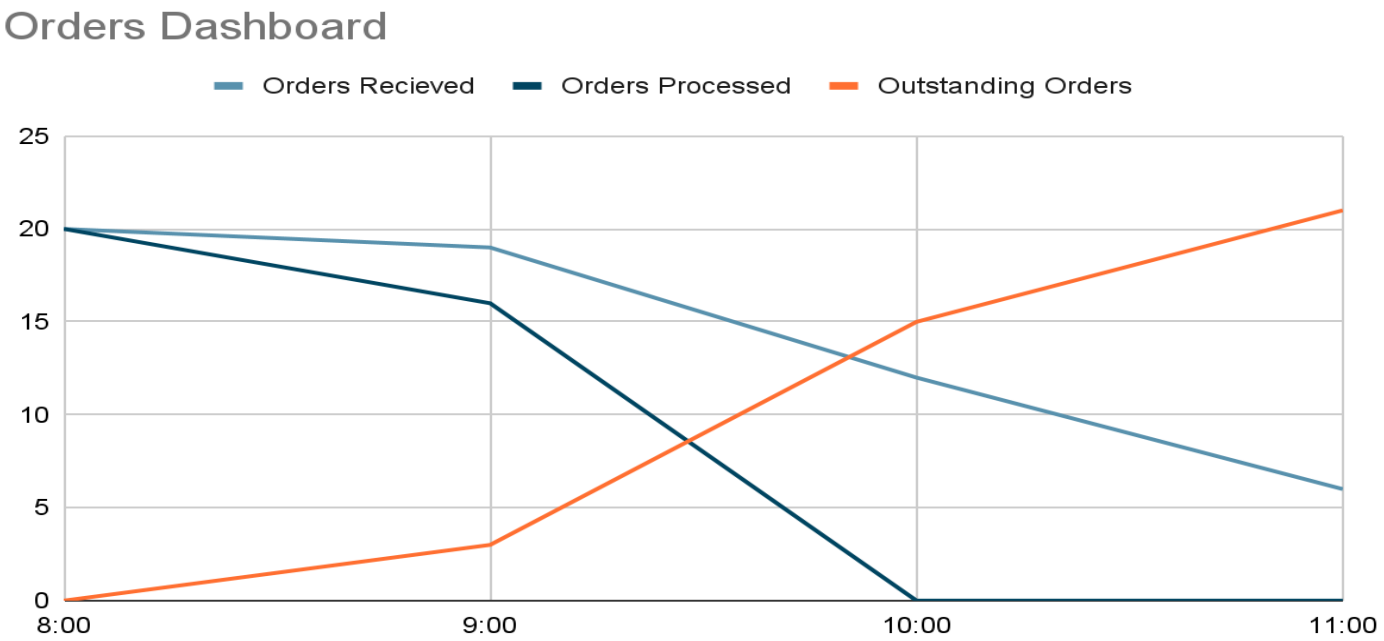
There are logs at /home/sre/course4/order_processing.log. If the server is not overloaded, this is a good place to check for errors. If you encounter any errors, send a message to the developers so that they can troubleshoot.

It is okay to restart this server during business hours. The Order Processor will pick up where it left off after a restart.

On-Call Shift -- Alert 1

Order Processing Issues, cont

Log/Monitoring Details




```
Order Processed
Processing Order 12
Order Processed
Processing Order 13
Order Processed
Processing Order 14
Order Processed
Processing Order 15
Order Processed
Processing Order 16
Order Processed
Processing Order 17
Order Processed
Processing Order 18
Order Processed
Processing Order 19
Order Processed
Processing Order 20
Order Processed
All Orders processed.

Startup...
Starting to process orders.
Processing Order 1
Order Processed
Processing Order 2
Order Processed
Processing Order 3
Order Processed
Processing Order 4
Error Processing Order. Error #12
Error Processing Order. Error #12
Error Processing Order. Error #12
Error Processing Order. Error #12
Error Processing Order. Error #12
Error Processing Order. Error #12
```

On-Call Shift -- Alert 2
Low Storage Alert

Summary

You receive an alert that the storage is running out on the mount where application logs are being written to. After consulting the SOP, you reach out to the team responsible for the server. They respond that Steve is normally in charge of handling the logs. Every morning he would run the commands listed in the run book, but he has been out sick for a week. The other members of the team forgot that it needed to be done, so the mount filled up.

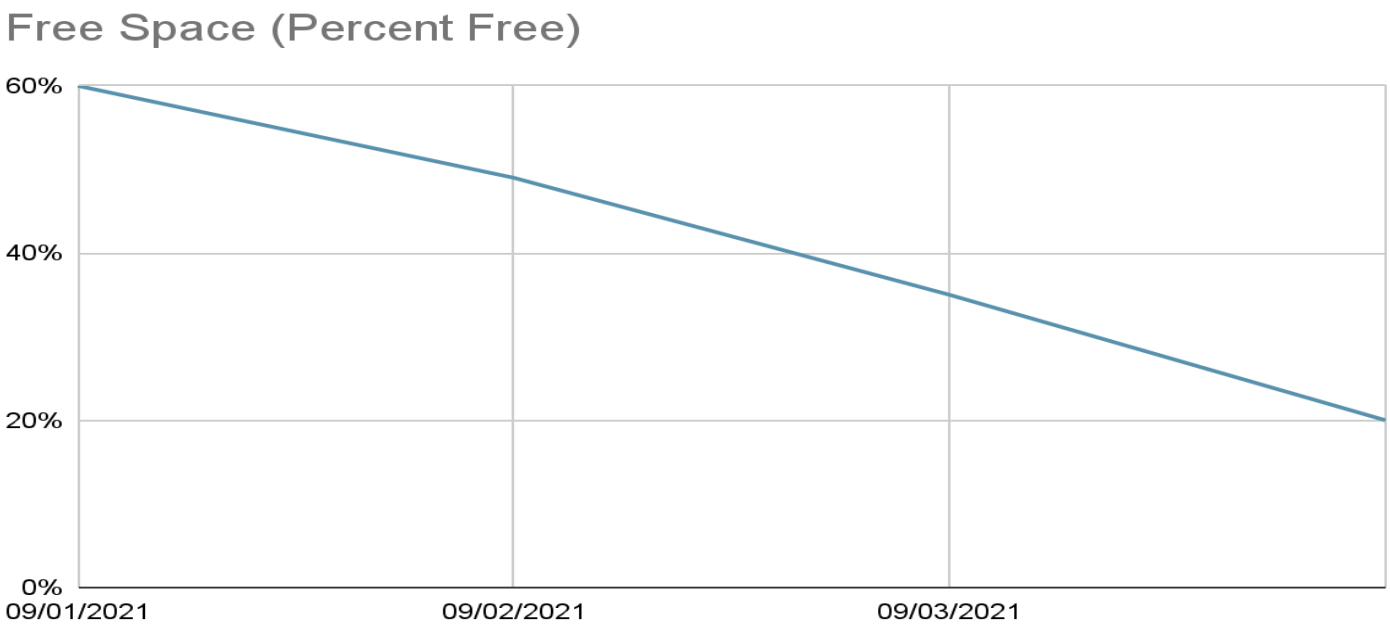
SOP

Low Storage

Depending on the specific alert take the following action:

/home/sre/course4/app.log -- If this mount is low on storage, reach out to Compliance. They will know what logs can be cleared out or will request additional storage.

Log/Monitoring Details



On-Call Shift -- Alert 3
DNS Troubles

Summary

The networking team recently added a secondary backup DNS server to increase reliability since the one they are using now tends to go down frequently. Your team has several checks in place monitoring the DNS servers to make sure they are up at all times.

SOP

DNS Server Not Answering Requests

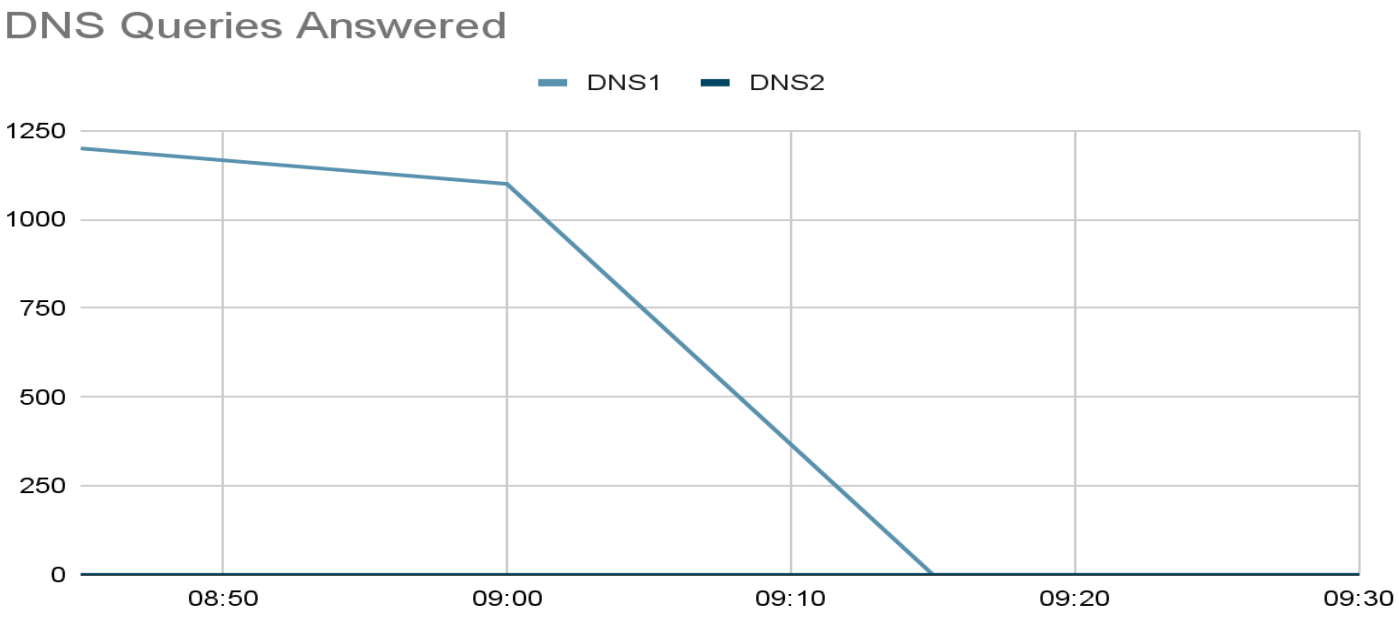
If you receive this alert, you should check to see if DNS1 or DNS2 is the current server answering requests. After determining which is the active server, check to see if the server is reachable. If the server is not reachable, immediately initiate the failover procedure to prevent any further network disruptions. If the server is reachable, check the logs to determine what the error is. If the active server cannot be brought back online within 5 mins, initiate the failover procedure. Either way, engage the Networking team to bring the standby server back online.

Failover Procedure

1. Determine the active server with the dnsTool.
    a. `dnsTool -q active_server`
2. If the active server is reachable you can initiate the shutdown process. If this command fails, make sure the dns process is shutdown on the server before continuing
    a. `dnsTool -a shutdown -s dns1`
3. Start the failover.
    a. If shutdown was successful:
    `dnsTool -a failover -s dns2`
    b. If shutdown was not successful, include the force flag,
    `dnsTool -a failover -s dns -f`

On-Call Shift -- Alert 3
DNS Troubles, cont

Log/Monitoring Details



Networking Server Status Page	
Server	Status
DNS1	UP
DNS2	UP

On-Call Shift -- Alert 3

DNS Troubles, cont

Log/Monitoring Details

```
Answering Query.
Answering Query.
Answering Query.
Answering Query.
Answering Query.
Answering Query.
Answering Query.
Answering Query.
Answering Query.
Answering Query.
Answering Query.
Answering Query.
Answering Query.
Answering Query.
Unexpected Error encountered.
Unexpected Error encountered.
Unexpected Error encountered.
Unexpected Error encountered.
Unexpected Error encountered.
Unexpected Error encountered.
Unexpected Error encountered.
Unexpected Error encountered.
Unexpected Error encountered.
```

On-Call Shift -- Alert 4
Application Outage

Summary

You receive the dreaded Application Down alert. Not only do you receive an alert for the application being down, but Customer Support also sent out a page to get all hands on deck for a report of the application being down.

SOP

Application Down

If you receive this alert, you need to act immediately. First, verify the application is indeed unreachable. If the application is unreachable, check to make sure the hosts are up and the application processes are running. You must start escalation for this immediately after verification the app is unreachable. Contact the following POCs:

Customer Support -- Susan Vega
Networking -- Bob Sparrow
Ops -- Glen Hammer
Database Admin -- Karen House
Development Team – Gal Tree

Log/Monitoring Details

Main App Status	
Endpoint or Host	Status
exoticplant.plant	UNREACHABLE
planthost1.internal	UP
planthost2.internal	UP
exoticplant.plant.internal	UNREACHABLE

On-Call Shift -- Alert 4

Application Outage, cont

Log/Monitoring, cont.

```
Info: Processing Request 407
Warming: Timeout. Retrying 428
Info: Processing Request 439
Warming: Timeout. Retrying 447
Warming: Timeout. Retrying 941
Warming: Timeout. Retrying 168
Warming: Timeout. Retrying 205
Warming: Timeout. Retrying 278
Info: Processing Request 439
Info: Placing Order 492
Warming: Timeout. Retrying 814
Info: Placing Order 520
Warming: Timeout. Retrying 662
Info: Processing Request 776
Info: Processing Request 548
Info: Processing Request 559
Warming: Timeout. Retrying 905
Info: Placing Order 948
Info: Placing Order 340
Error: Var is 10 RETRYING
```

On-Call Shift -- Alert 4

Application Outage, cont

Log/Monitoring, cont.

09:15 Hey we have reports of an application outage and we can not reach the app either. **FROM: svega**
09:16 I have an alert for that too. I'm looking at things now, will start a communication channel to coordinate. Checking logs and app servers now. **FROM: YOU**

09:20 -- !svega !bsparrow !ghammer !khouse !gtree we have an application outage **FROM: YOU**.
0930 -- Everything looks good from the network **FROM: sparrow**
0932 -- I can access the DB and it is reporting back normal **FROM: khouse**
0935 -- Everything here looks normal. FROM: ghammer
0937 -- We are still reviewing logs and seeing if we can reproduce on our end FROM: gtree
0938 -- We should try restarting the app, Maybe that will help FROM: ghammer
0940 -- Maybe that will help. FROM: svega
0943 -- Okay I will try. Bringing down. FROM: YOU
0945 -- App is down. Bring back up. FROM: YOU
0947 -- App is starting. FROM: YOU
0952 -- Main app is back up. FROM: hammer
0955 -- App is still not respond. FROM: svega
0956 -- I'm sending you some new logs !gtree these look off FROM: hammer
1005 -- !sre !ghammer when was the last deploy? What were the details? This looks like a qa build. FROM: gtree
1007 -- I did a deploy with one of the devs to qa to do some testing. Let me check. FROM: ghammer
1010 -- I think there was a mixup when doing the deployment. The wrong scripts was used and that build was deployed to prod. FROM hammer
1011 -- Were there any migrations for that !ghammer FROM: khouse
1012 -- No, just code changes. FROM: hammer
1013 -- Thats good. We should be able to just revert back then. !svega
1015 -- Let me take down the app and redeploy it. FROM: YOU
1017 -- App is down. Bring back up. FROM: YOU
1023 -- App is starting. FROM: YOU
1026 -- Main app is back up. FROM: hammer
1030 -- Everything looks like it is responding now. FROM: svega

On-Call Summary Log Template

Date/10:00 -- *Alert 1.* Order Processing Issues

Troubleshooting

- I receive an alert that the number of Outstanding Orders is too high.
- Orders are processed by a separate component from the main application. It runs periodically (every hour currently) to batch process any open orders.
- My team has set up some monitors to keep track of how well the order processor is doing.
- We follow the SOP procedure:

1. Observing the Orders Dashboard Graph, the number of outstanding orders has increased sharply from 3 outstanding orders (at 9:00 am) to 15 outstanding orders (at 10:00 am). Also, we note that the number of orders processed decreased decreased, at 9:00 am there are 16 order processed while at 10:00 am there are 0 order processed. So we move on to next point of procedure SOP.
2. We go to the register folder at the address: /home/sre/course4/order_processing.log. As the server is not overloaded, it is verified that error of the type are observed : “Error Processing Order. Error #12” corresponding to at 10:00 am batch. We sent a message to the developers to fix the problem.
3. The server restarts during business hours. The Order Processor is shown to pick up where it left off after the restart.

Resolution

We solved the alert of too high of outstanding orders numbers by:

1. I send a message to the developer group to fix the problem that errors are observed like: “Error Processing Order. Error #12”.

2. The server restarts during business hours. The Order Processor is shown to pick up where it left off after the restart.

On-Call Summary Log Template

09.03.2021/08:00 -- *Alert 2. Low Storage Alert*

Troubleshooting

- I received an alert that the storage is running out on the mount where application logs are being written to. After consulting the SOP, I reach out to the team responsible for the server. They respond that Steve is normally in charge of handling the logs. Every morning he would run the commands listed in the run book, but he has been out sick for a week. The other members of the team forgot that it needed to be done, so the mount filled up.
- In the Free Space I can see from 9 January 2021 until 9 April 2021 that is observed a reduction in the free space around 40%. In 09/03/2021 we have a 35% of free space.
- I followed the procedure SOP, I the case of low storage I do the next step:
    1. I wet to /home/sre/course4/app.log — like this mount is low on storage, I reach out to Compliance. They will know what logs can be cleared out of will request additional storage

Resolution

I went to /home/sre/course4/app.log — like this mount is low on storage, I reach out to Compliance. They will know what logs can be cleared out of will request additional storage.

On-Call Summary Log Template

09.03.2021/08:00 -- *Alert 2. Low Storage Alert*

Troubleshooting

- I received an alert that the storage is running out on the mount where application logs are being written to. After consulting the SOP, I reach out to the team responsible for the server. They respond that Steve is normally in charge of handling the logs. Every morning he would run the commands listed in the run book, but he has been out sick for a week. The other members of the team forgot that it needed to be done, so the mount filled up.
- In the Free Space I can see from 9 January 2021 until 9 April 2021 that is observed a reduction in the free space around 40%. In 09/03/2021 we have a 35% of free space.
- I followed the procedure SOP, I the case of low storage I do the next step:
    1. I went to /home/sre/course4/app.log — like this mount is low on storage, I reach out to Compliance. They will know what logs can be cleared out of will request additional storage

Resolution

I went to /home/sre/course4/app.log — like this mount is low on storage, I reach out to Compliance. They will know what logs can be cleared out of will request additional storage.

On-Call Summary Log Template

Date/09:15 -- *Alert 3.* DNS Troubles

Troubleshooting

- The networking team recently added a secondary backup DNS server to increase reliability since the one they are using now tends to go down frequently. I started with my team to have several checks in place monitoring the DNS servers to make sure thy are up at all times.
- In the monitoring system in the DNS Queries Answered we had an alert at 9:15 that the DNS1 not answering requests.
- Next step was to follow the DNS Server Not Answering Request SOP:

1. We checked that the DNS1 and DNS2 are reachable, in the networking server status page we observe that the DNS1 and DNS2 are with status up.
2. How the servers are reachable, we checks the logs to determine what the error is. In the logs we observed that the DNS1 has the follow fail: “Unexpected Error encountered”. The active server couldn’t be brought back online in 5 minutes, we initiated the failover procedure.
3. Failover Procedure
    1. Determine the active server with the dnsTool.
        - dnsTool -q active_server
    2. If the active server is reachable you can initiate the shutdown process. If this command fails, make sure the dns process is shutdown on the server before continuing
        - *dnsTool -a shutdown -s dns1*
    3. Start the failover.
        - If shutdown was successful:
          dnsTool -a failover -s dns2
        - If shutdown was not successful, include the force flag, *dnsTool -a failover -s dns -f*

Resolution

- We follow the Failover Procedure because the status of both DNS servers is UP and the DNS 1 that is the main DNS server has logs with the fail: “Unexpected Error encountered”. The active server couldn’t be brought back online in 5 minutes, we initiated the failover procedure.

On-Call Summary Log Template

Date/09:15 -- *Alert 3.* DNS Troubles

- Failover Procedure

    1. Determine the active server with the dnsTool.

        - dnsTool -q active_server

    2. If the active server is reachable you can initiate the shutdown process. If this command fails, make sure the dns process is shutdown on the server before continuing

        - *dnsTool -a shutdown -s dns1*

    3. Start the failover.

        - If shutdown was successful:
          dnsTool -a failover -s dns2

        - If shutdown was not successful, include the force flag, *dnsTool -a failover -s dns -f*

On-Call Summary Log Template

Date/09:15 -- *Alert 4. Application Outage*

Troubleshooting

- At 9.05 am I received the dreaded Application Down alert from Susan Vega (Customer Support). Not only do I receive an alert for the application being down, but Customer Support also sent out a page to get all hands on deck for a report of the application being down.

- I follow the SOP Application Down procedure:

  1. SRE team verified that the application is indeed unreachable:

     A. We verified the monitoring details of endpoint and host status. We saw that exotic pant.plat and exotic plant.plant.internal are unreachable.

     B. We verified the logs and we saw the follow fail: “Error: Var is 10 retrying”. Also we saw that request are in the state: “Warming:Timeout. Retrying ”

  2. SRE team verified that the application was shutdown and I started escalation for this immediately after verification the app is unreachable. I have contacted with the following POCs:

     Customer Support -- Susan Vega
     Networking -- Bob Sparrow
     Ops -- Glen Hammer
     Database Admin -- Karen House
     Development Team – Gal Tree

- We tried to recover the service for 2 ways:

  1. The first attempt to recover the service was:

     A. SRE team restart the Application and Customer Support verified that the service is unreachable.

  2. The second attempt to recover the service was:

     A. At 10:11 there were a conversation between Ops and Database Admin. Ops where asked to Database Admin were there any migration?. At 10:12 Ops answered that there were some changes in the code.

     B. At 10:15 the SRE team download decided to take down and redeploy the app. At 10:30 Customer support that the app is reachable.

On-Call Summary Log Template

Date/09:15 -- *Alert 4. Application Outage*

Resolution

- The problem was that the Ops team did some code changes in the code of the application. The solution was that the SRE team take down the application and redeploy the code with the success result.

Post-Mortem Template

Incident Title -- Date/Time

Stakeholders

This should include all teams and individuals who were involved in the incident.

Incident Timeline

This is a timeline of the events from when the incident was reported to resolution. Make sure to include both events by actual persons (Joe logged on to server1 and restarted the service) as well as system events (From the logs in server2, we see that network connectivity stopped at 14:32).

Impact

This section should include an impact assessment that describes how the business, customers, and systems were affected. The outage affected the order processing system preventing orders from being processed. This led to customers having delayed orders, as well as having to pull additional business resources in to process orders manually. This led to a loss of revenue for the business.

Resolution

Describe what was specifically was done to resolve the issue. This section can be used to document scripts, commands, actions, or vendor support engaged that may be useful for follow-up or automation.

Action Plan

This will be a plan of action to prevent the incident from reoccurring. It should include any safeguards to be implemented, automation to be performed, additional redundancy to be added, etc. This should also include a breakdown of who will perform what and when it should be implemented by.

Post-Mortem

Application Outage -- Date/09:15

Stakeholders

- Customer Support
  - Susan Vega
- Networking
  - Bob Sparrow
- Ops
  - Glenn Hammer
- Database Admin
  - Karen House
- Development Team
  - Gal Tree
- SRE
  - César Blanco Fernández

Incident Timeline

09:15: Susan Vega from Customer Support reporte that an application outage
“Hey we have reports of an application outage and we can not reach the app either. **FROM: svega”**
09:16: César Blanco Fernández from SRE Team check in the monitoring that he has an alert. He followed the SOP procedure and he started a communication channel to coordinate different groups: Susan Vega (from Customer Support), Bob Sparrow (from Networking), Glenn Hammer (from Ops), Karen House (from Database Admin) and César Blanco Fernández (from SRE).
“I have an alert for that too. I’m looking at things now, will start a communication channel to coordinate. Checking logs and app servers now. **FROM: YOU”**
09:30 : Bob Sparrow reported that network is ok
“Everything looks good from the network **FROM: sparrow”**

Post-Mortem

Application Outage -- Date/09:15

Incident Timeline (continue)

09:32 -- Karen House report that she can access to the DB and it is reporting back normal I can access the DB and it is reporting back normal **FROM: khouse**
09:35 — Glenn Hammer report that in the Ops part of the Application is normal
“Everything here looks normal. FROM: ghammer”
09:37 -- Gal Tree report that he is still reviewing logs and seeing if we can reproduce on our end
“We are still reviewing logs and seeing if we can reproduce on our end FROM: gtree”
09:38 — Glenn Hammer suggest to restart the application
“We should try restarting the app, Maybe that will help FROM: ghammer”
09:40 — Susan Vega agrees to restart the application
“Maybe that will help. FROM: svega”
Between 09:43 to 09:47 César Blanco Fernández proceed with the restart process
“09:43 -- Okay I will try. Bringing down. FROM: YOU
09:45 -- App is down. Bring back up. FROM: YOU
09:47 -- App is starting. FROM: YOU”
09:52 -- Glenn Hammer report that main app is back up
“Main app is back up. FROM: hammer”
09:55 -- Susan Vega report that main application is unreachable
“App is still not respond. FROM: svega"
09:56 -- Glenn Hammer send new logs to Gal Tree.
“I’m sending you some new logs !gtree these look off FROM: hammer”
10:05 -- Gal Tree ask to César Blanco and Glenn Hammer: when was the last deploy? What were the details?. He says that this looks a qa build.
“!sre !ghammer when was the last deploy? What were the details? This looks like a qa build. FROM: gtree”
10:07 -- Glenn Hammer answered that he did a deploy with one of the devs to qa to do some testing.
“I did a deploy with one of the devs to qa to do some testing. Let me check. FROM: ghammer”
10:10 — Glenn Hammer suggest that he think there were a mixup when doing the deployment. He suggest that the wrong scripts was used and that build was deployed to production.
“I think there was a mixup when doing the deployment. The wrong scripts was used and that build was deployed to prod. FROM hammer”

Post-Mortem

Application Outage -- Date/09:15

Incident Timeline (continue)

10:11 -- Karen House ask to Glenn Hammer if there were any migration
“Were there any migrations for that !ghammer FROM: khouse”
10:12 -- Glenn Hammer answer no and he changes only code
No, just code changes. FROM: hammer
10:13 — Susan Vega is happy because we should be able to revert back
“Thats good. We should be able to just revert back then. !svega”
Between 1015 to 10:23 César Blanco takes down, redeploy and start the application
“10:15 -- Let me take down the app and redeploy it. FROM: YOU
10:17 -- App is down. Bring back up. FROM: YOU
10:23 -- App is starting. FROM: YOU”
10:26 -- Glenn Hammer confirm that main app is back up
“Main app is back up. FROM: hammer”
10:30 -- Susan Vega confirm that main app is up too
“Everything looks like it is responding now. FROM: svega”

Post-Mortem

Application Outage -- Date/09:15

Impact

This outage has an impact in different assessment like:

A. Business: the duration of outage was 1hour and 15 minutes. In this time the web couldn't process order and this thing has an impact in loss of revenues for the business.

B. Customer: In the time that the customer couldn't buy in the e-commerce and process the order in the system. On the other hand, the outage damage the image of the e-commerce and the client could leave to buy in the e-commerce and not recommend it.

C. System: The outage affected the order processing system preventing orders from being processed.

Resolution

- The problem was that the Ops team did some code changes in the code of the application. The solution was that the SRE team take down the application and redeploy the code with the success result.

Post-Mortem

Application Outage -- Date/09:15

Action Plan

The action plan to avoid another outage in the application. We could choose one of these plan:

- Plan 1

1. Migrate the service to cloud because give us: cost saving, security, flexibility, quality control, disaster recovery, loss prevention, automatic software updates.

2. Add in the system architecture the follows components and devices:

    A. Deploy the architecture in a Cloud Region

    B. Deploy every component in a VPC (Virtual Private Cloud)

    C. Load Balancer Balancer to share the load to different virtual machines

    D. Virtual Machines that include a Kubernetese cluster with micro-service for the main application and order processor.

    E. Implement an auto-scaling group, to have an increase of virtual machines (scaling out) or decrease (scaling in) of the virtual machines with the load of the server.

    F. Add a Primary Data Base in an availability zone and another Replica Data Base in another availability zone to release the load of the order processor with the outstanding orders.

3. Implement a Disaster Recovery Plan with the goal to have a zero downtime. An indicative diagram can be as follows. (Fig. 1. Pilot light DR strategy). To make that, we will need to replicate all the infrastructure in another region but in this case there are only a Replica Data Base.

4. Automation all task to reduce the toil:

    A. We use Terraform software to automation deploy of all components of architecture.

    B. Using run book with the scripts to deploy the software.

Post-Mortem

Application Outage -- Date/09:15

Action Plan

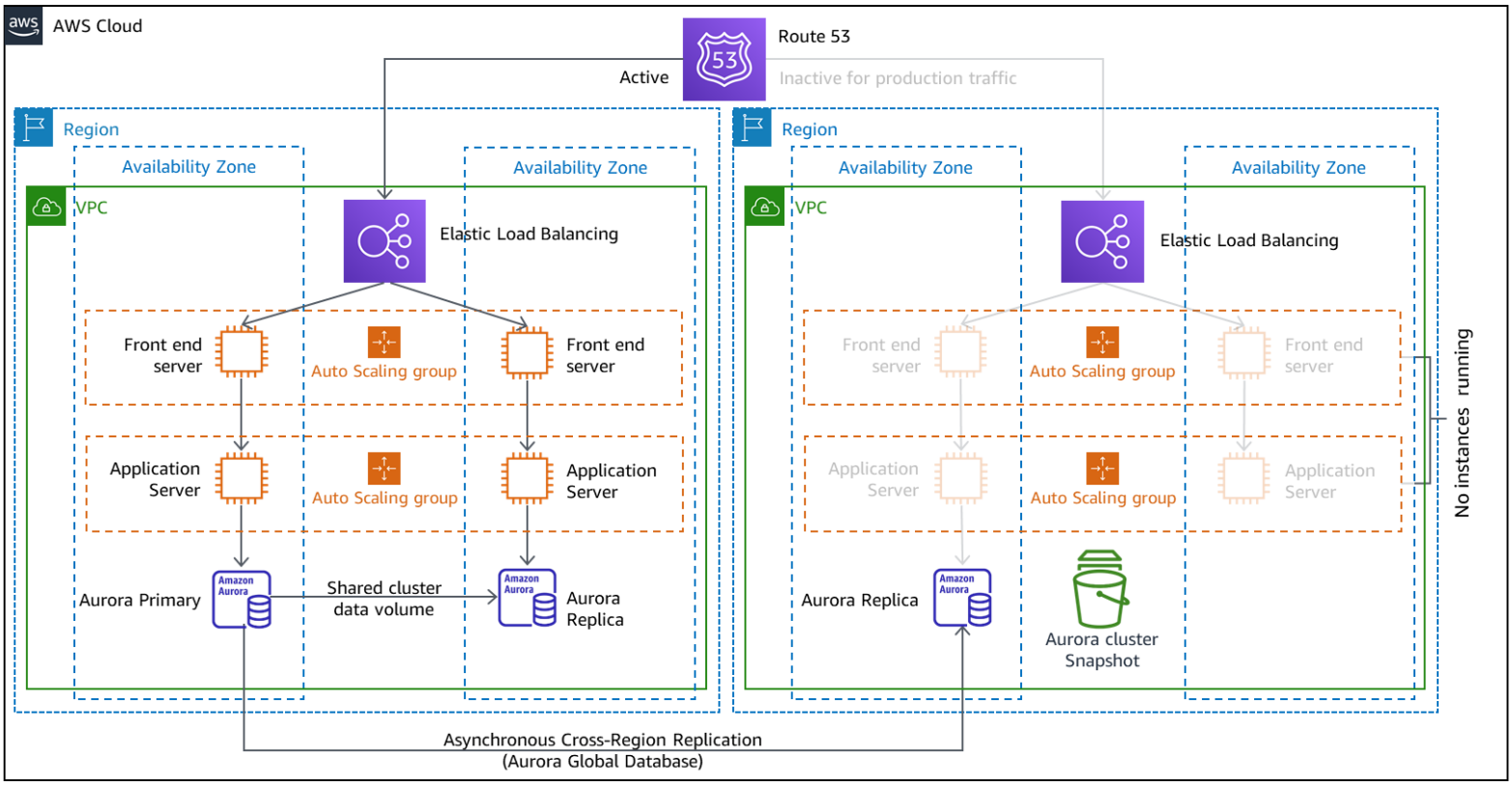


Fig. 1. Pilot light DR strategy

- To avoid that another change in code leave an outage in the application, it is necessary to test the code in environment like our environment of the main application.
- We could use different strategies to deploy the new version code to avoid the outage:

    A. Rolling Deployment

    B. Canary Deployment

    C. Blue Green Deployment

Post-Mortem

Application Outage -- Date/09:15

Action Plan

- Plan 2

We continue with a deploy on-premise infrastructure.

1. Add in the system architecture the follows components and devices:

    A. Load Balancer Balancer to share the load to different Virtual Machines with the main application and order processor.

    B. Virtual Machines that include a Kubernetes cluster with micro-service for the main application and order processor.

    C. Implement an auto-scaling group, to have an increase of virtual machines (scaling out) or decrease (scaling in) of the virtual machines with the load of the server.

    D. Add a Primary Data Base and another Replica Data Base to release the request of the order processor in the main data base with the outstanding orders.

2. Automation all task to reduce the toil:

    A. We use Terraform software to automation deploy of all components of architecture.

    B. Using run book with the scripts to deploy the software.

3. We could use different strategies to deploy the new version code to avoid the outage:

    A. Rolling Deployment

    B. Canary Deployment

    C. Blue Green Deployment

Scenario 3

Toil Reduction

Toil Reduction Plan

Summary

Now that you have spent some time on your own as an SRE, you now have to round out your week by handling some of the toil you encountered. Looking through the on-call summary, post-mortem, as-built design doc, and your experience, you decided that there are several ways to reduce toil. You start by listing out 5 of the major items for this week. For each one, you analyze the impact on the business and what you gain by automating the task. After that, you will need to implement three of these items in pseudocode to help your team move forward.

Current Toil Items

Item 1: Manually configuring new servers (physical or virtual) with standard set configuration

Manually configuring a new servers requiere a lot of toil. This is a repetitive task that we could automate with a script.

With this automation solution we are saving time in repetitive task and the technicals can be devoted to other tasks.

Item 2: Restarting main application

This a task that in Application outage we had to restart the service sometimes to recover the service in the e-shop. That is a automatable and we could using a script. With this script we have the guarantee of using all the necessary commands and not making any mistakes when restarting the application.

With this automation solution we are saving time in repetitive task and the technicals can be devoted to other tasks.

Toil Reduction Plan

Item 3: Failover procedure DNS

We could reduce the toil to do the failover procedure to hit manually the commands.

The solution for this problem is the automation of this task with a script which facilitates the management of procedure.

With this solution we get time saving in management tasks for the technicians that can be devoted to other eventualities.

Item 4: System Resource Management

In the actual architecture system existing servers are physical and it is very hard to plan and manage capacity.

The solution to this problem is doing a capacity planning, and we exchange this physical server with VMs. The Virtual Machines are easier to add resources as necessary.

The impact in the business is low because the cost to deploy these components is low.

Item 5: System Configuration

Actually, we configure the system the manual way. With that, there is a risk that technicians make a fail when develop a repetitive with command.

The solution is the Automation.Using a system configuration tool to automate configuring systems saves times and keeps systems time and keeps systems from drifting too far from their configs.

Item 6: Low Storage

Actually we detected the storage space with a graph without any threshold and trigger to review the alert and free up the mount. Every check of free space is manually.

To avoid the problem, that we had with the alert of free space in the Storage, the solution for this problem is Automation. We write a script with 2 action to do:

1. Alert low storage with an trigger alert

2. Free up storage

3. Alternative send an e-mail alert

Toil Reduction Plan

Item 6: Low Storage (continue)

With this solution we get time saving in management tasks for the technicians that can be devoted to other eventualities.

The impact in the business is high because we could solve the problem of low storage space before having trouble storing order management customers data.

Automation Implementation

Item 2: Restarting main application

Script to restart main application with name ***restart_main_app.sh***:

#!/bin/bash

killall -I -q process-name

sleep 2m

start-command

Note:

- Replace process-name with the name of the process.
- Replace start-command with the command to start the process.

Ref.https://gist.github.com/SYZYGY-DEV333/e82d45d5a4cdfecf3602

Automation Implementation

Item 3: Failover Procedure DNS

Script to failover Procedure DNS with name ***failover_procedure_dns.sh***:

```
#!/bin/bash

if  dnsTool -q active_server |  grep -q 'active' ; then

    echo ' dns 1 server is reachable'
    sleep 5

    #start the failover to dns2 server
    dnsTool -a failover -s dns2
    sleep 2m
    echo 'success failover process'

else
    echo ' dns 1 server is unreachable'
    sleep 5

    #shutdown dns1 server
    dnsTool -a shutdown -s dns1

    #start the failover to dns2 server
    dnsTool -a failover -s dns2
    sleep 2m
    echo 'success failover process'

fi
```

Ref. :
https://stackoverflow.com/questions/16931244/checking-if-output-of-a-command-contains-a-certain-string-in-a-shell-script

Automation Implementation

Item 6: Low Storage

Script to failover Low Storage with name ***free_space.sh***:

```
#!/bin/sh
# Purpose: Monitor Linux disk space, send an email alert to $ADMIN
# and free up storage space

#Variables
ALERT=70 # alert level
ADMIN="ceblfe@gmail.com" # dev/sysadmin email ID
TMP_DIR="/tmp"

#Function to clear cache
function clear_cache {

    #Calculates the current percentage of free RAM available in system
    _ram_avl=`free | grep Mem | awk '{print $4/$2 * 100.0}' | awk -F'.' '{print $1}'`

    #If free percentage is less than equal to 15% of total RAM trigger the cleanup

    if [ $_ram_avl -le 15 ]; then

    echo "RAM available in system is below 15 percentage clearing cache and freeing up
memory";
    echo
    free && sync && echo 3 > /proc/sys/vm/drop_caches && free
    echo
    sudo sysctl vm.drop_caches=1
    echo
    fi;
}


# 1. get disk space ,
#    filter out filesystem and
#    find out the percentage of space
df -H | grep -vE '^Filesystem|tmpfs|cdrom' | awk '{ print $5 " " $1 }' | while read -r output;
```

Automation Implementation

```
do
  echo "$output"
  usep=$(echo "$output" | awk '{ print $1}' | cut -d'%' -f1 )
  partition=$(echo "$output" | awk '{ print $2 }' )
  if [ $usep -ge $ALERT ]; then

    #2. Send one e-mail alert with low storage
    echo "Running out of space \"$partition ($usep%)\" on $(hostname) as on $(date)" | mail -s
"Alert: Almost out of disk space $usep%" "$ADMIN"
    sleep 5

    #3. Remove unused dependencies
    apt-get autoremove
    sleep 10

    #4. Removing temporary files
    echo "Removing all temporary files from $TMP_DIR"
    trap '{ rm -f -- "$TMP_DIR"; }' EXIT

    # Counting the number of temporary files
    files=`ls -l $TMP_DIR | wc -l`

    echo "There are total $files temporary files/directory in $TMP_DIR directory"
    sleep 5

    #5. Clear cache
    clear_cache
    echo "Cache is cleared"

  fi
done
```

Ref:
https://www.baeldung.com/linux/clean-up-linux-system

https://www.techpaste.com/2018/02/shell-script-linux-clear-cache/

https://stackoverflow.com/questions/687014/removing-created-temp-files-in-unexpected-bash-exit

https://www.geeksforgeeks.org/shell-script-to-remove-temporary-files/