Project:
Plan, Reduce, Repeat

[Dientv]:
[06/06/2023-v1]

How to Use this Template

- Make a copy of this Google Slide deck.
- We have provided these slides as a guide to ensure that you submit all the required components to successfully complete your project.
- When presenting your project, please only think of this as a guide. We encourage you to use creative freedom when making changes as long as the required information is present.
- Feel free to create additional slides if you need more space to write your responses or include screenshots.
- **Feel free to delete this page and the other pages with instructions** before you submit your project.

Reference slide remove before you submit

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 usage by 25% for the same number of users while not increasing CPU significantly. Currently, the main application containers utilize almost 85% of the RAM allocated.
  - At the current resource allocation, each replication can handle 500 concurrent users. Currently, there are 3 application containers to support 1500 total users. This release is expected to add about 2.5 times the total number of users.
- Order Processor
  - This component has a high CPU utilization with moderate RAM requirements. In testing, a fully loaded queue used a bit less than 1 Gb of RAM.
  - The component runs with 4 concurrent processes, pulling orders from the database and processing them for fulfillment. 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
  - Jane 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

Stakeholders

- Developers
  - Dien Trinh
  - Hoang Vo
  - Huy Anh
- Ops
  - Chien Tran
- SRE
  - Quan Hoang

Code Changes

- Improvement features
  - Decoupling app to improve the performance(Tk-106)
- Feature Additions
  - Added batch processing components(Tk-107)

Data and System Changes

- Data model changes
  - Added table for batch processing (TK-109)
  - Added additional columns to order table to support batch processing (Tk-107)

As-Built Doc
Release 2

Design decision highlights

Add batch processing for order processing. Decoupling order application to improve performace

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 2

```
ApiVersion: apps/v1
kind: Deployment
metadata:
  name: app-deployment
  namespace: course4
  labels:
    app: mainApp
spec:
  Replicas: 8
  selector:
    matchLabels:
      app: mainApp
  template:
    metadata:
      labels:
        app: mainApp
    spec:
      containers:
      - name: mainApp
        image: nginx:latest
        resources:
          requests:
            memory: 320mb
            cpu: 250m
        ports:
        - containerPort: 80
      - name: order_processor
        image: nginx:latest
        resources:
          requests:
            memory: 512mb
            cpu: 1Gi
        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

Low Storage Alert

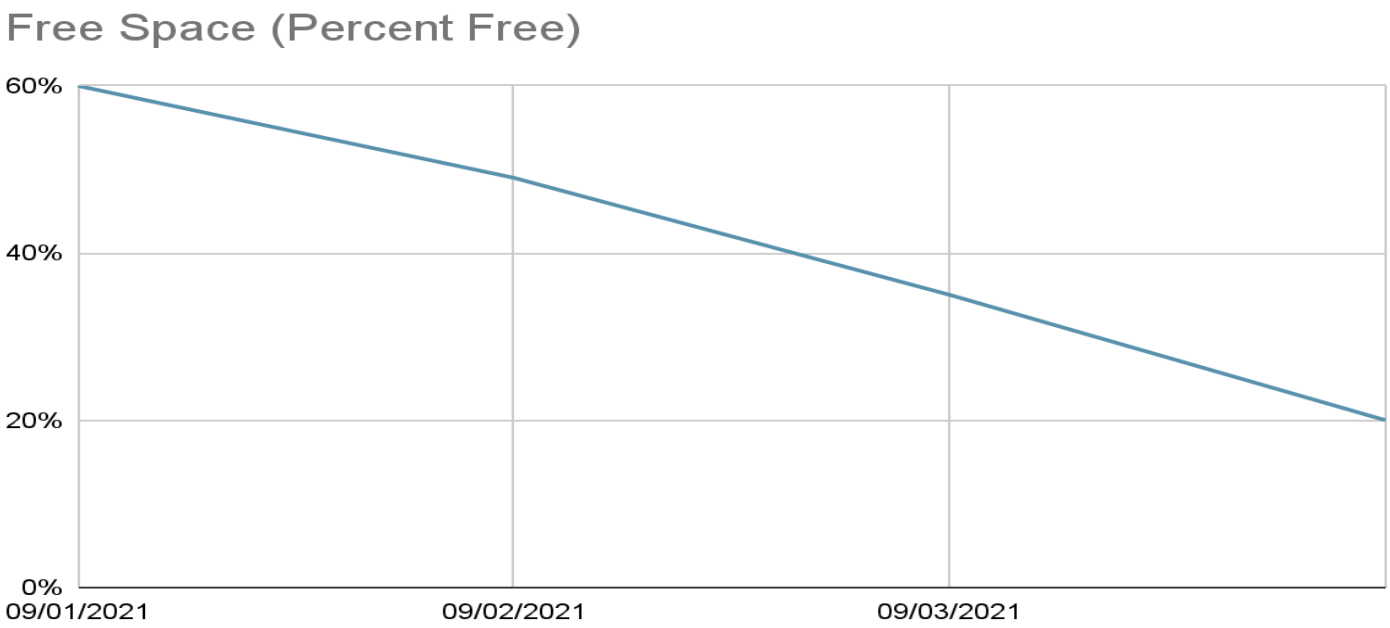
Summary

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 2

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.

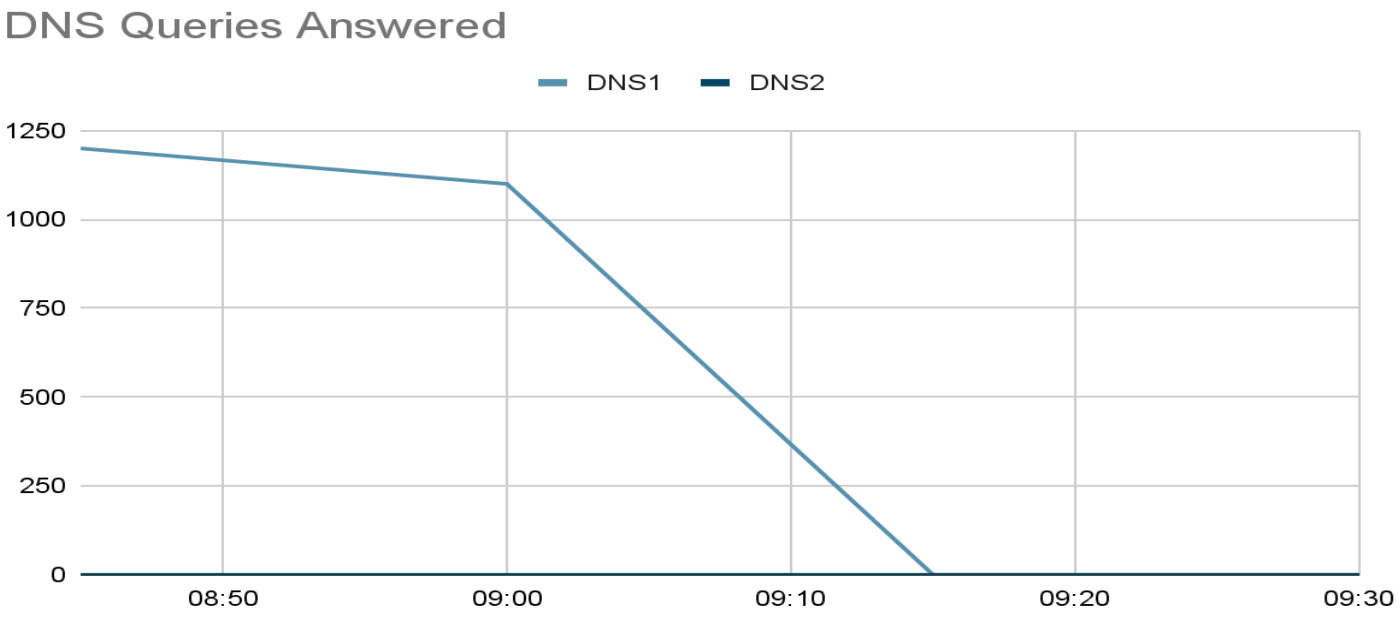
Failover Procedure

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 2

DNS Troubles, cont

Log/Monitoring Details



Networking Server Status Page	
Server	Status
DNS1	UP
DNS2	UP

On-Call Shift -- Alert 2

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 3

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 3

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 3

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/Time -- *Alert 1*

Troubleshooting

- [ ] Went to our monitoring stack to check if storage is running out on the mount storage.
- [ ] Checked the log I saw that currently the storage space is low. It is 20% free only.
- [ ] Sent an email to Compliance team to report the issue so they can handle.

Resolution

Reach out to Compliance team so that they will handle to free up the space of the storage either clear out log or request additional storage.

On-Call Summary Log Template

Date/Time -- *Alert 2*

Troubleshooting

- [ ] Went to the DNS system to check if both DNS server are up and confirm that both server are up.
- [ ] Initiate the failover procedure to DNS1.
- [ ] Check the logs and see that the DNS server was able to answer the request from 0 -> 9:15 AM after that it was not able to answer the request with error "Unexpected Error Encounter".
- [ ] Engage the Networking team to bring the standby server back online

Resolution

Initiate the failover procedure to DNS1 and engage the Networking team to bring the standby server back online.

On-Call Summary Log Template

Date/Time -- *Alert 3*

Troubleshooting

- [ ] Received the alert from customer about the application is outage.
- [ ] Checked with network, ops and database teams to see if everything are normal from their side and they confirmed everything was ok from their side.
- [ ] Checked logs and saw that the application was not response.
- [ ] Restarted application but still not resolved the issue.
- [ ] Figured out the issue relating to the mix-up when doing the deployment so had to redeploy again and application was back.

Resolution

Redeploy the fixed version to make the application operations normally.

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 – June 3, 14h:30

Stakeholders

- Dien Trinh
- Duy Nguyen

Incident Timeline

- From June 3, 14h:00 - June 3, 15h:00

Impact

- Users were unable to process the order while service Order processing was unavailable from 14h:00 to 15h:00 UTC

Resolution

- Restart the server so that app will be up again to mitigate the problem.

Action Plan

- Check the log and metrics to know the root causes why application outage
- Update the terraform to enable the auto scaling and send notification if the RAM/CPU is reached the threasholds

Scenario 3

Toil Reduction

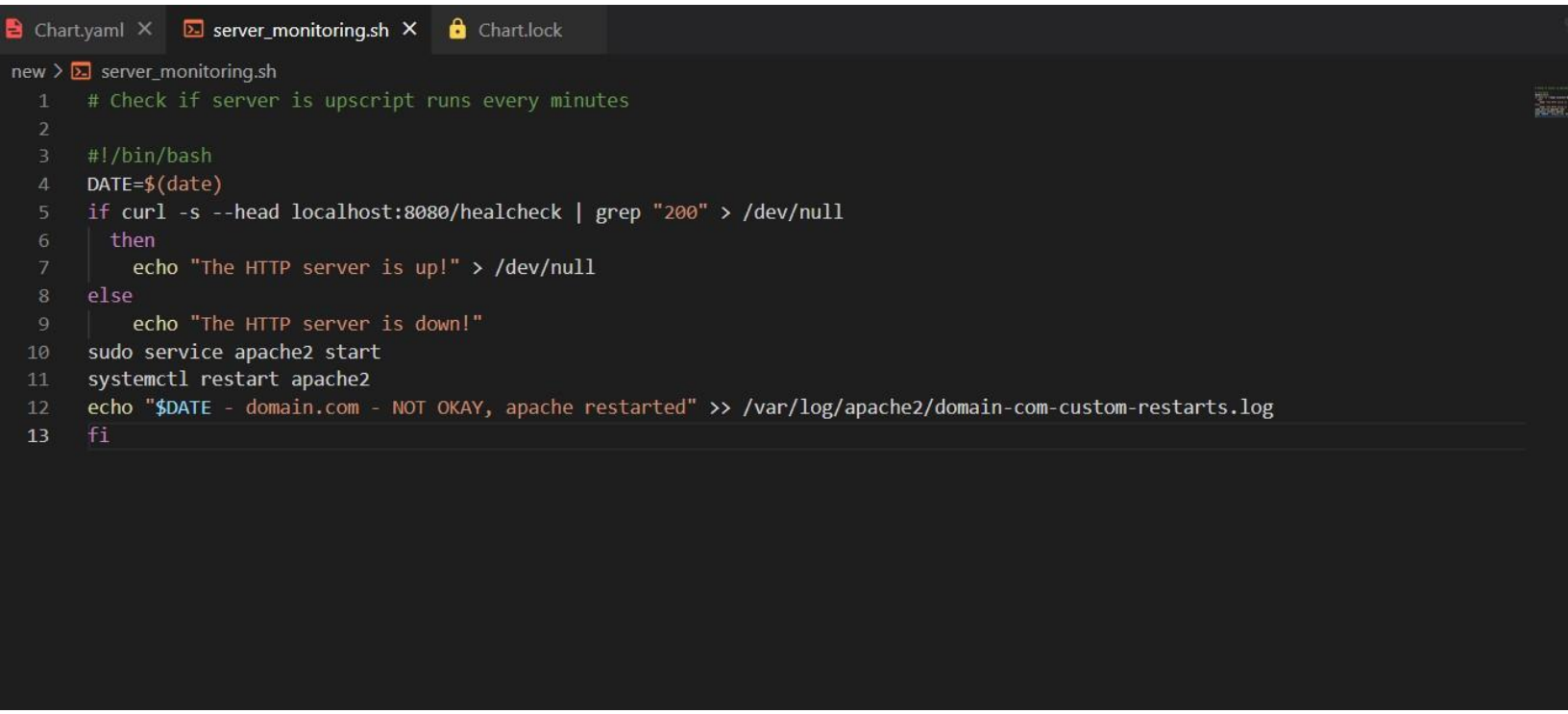
Toil Reduction Plan

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 need to list out 4 of the major items for this week, explain the reason for choosing each item briefly, and provide at least one benefit automating the toil would have. After that, you will need to implement two of these items in pseudocode to help your team move forward.

Toil Items	**Why it is considered toil?**	**Benefits of automating**
Backup database weekly	*It is occurred weekly, automatically and stateless*	*Reduce the resources, automatically, avoiding human error*
Backup server weekly	*It is occurred weekly, automatically and stateless*	*Reduce the resources, automatically, avoiding human error*
Restart application if RAM is 100% usage and not responding	Because it happens repeatedly if RAM is high and can use script to restart automatically instead of doing manually	Reduce the downtime if app down.
Clean unused files on tmp folder to save the disk space	During CI-CD process some files are created but not used afterward. It is consider as toil because it occurred frequently, stateless and can apply automatically	Save the space disk, save cost

Automation Implementation



Automation Implementation

tumlum > backup_db.sh

```
#!/bin/bash

# BEGIN CONFIGURATION =====================================================

BACKUP_DIR="/backups/"  # The directory in which you want backups placed
KEEP_MYSQL="14" # How many days worth of mysql dumps to keep

MYSQL_HOST="localhost"
MYSQL_USER="root"
MYSQL_PASS=""
MYSQL_BACKUP_DIR="$BACKUP_DIR/mysql/"

# You probably won't have to change these
THE_DATE="$(date '+%Y-%m-%d')"

MYSQL_PATH="$(which mysql)"
MYSQLDUMP_PATH="$(which mysqldump)"
FIND_PATH="$(which find)"
TAR_PATH="$(which tar)"
RSYNC_PATH="$(which rsync)"
# END CONFIGURATION =======================================================

# Announce the backup time
echo "Backup Started: $(date)"

# Create the backup dirs if they don't exist
if [[ ! -d $BACKUP_DIR ]]
  then
  mkdir -p "$BACKUP_DIR"
fi
if [[ ! -d $MYSQL_BACKUP_DIR ]]
  then
  mkdir -p "$MYSQL_BACKUP_DIR"
fi


# Get a list of mysql databases and dump them one by one
echo "-----------------------------------"
ALL_DBS=($($MYSQL_PATH -h $MYSQL_HOST -u$MYSQL_USER -p$MYSQL_PASS -Bse 'show databases'))
SYSTEM_DBS=("information_schema" "mysql" "performance_schema" "test")
DBS=()

for i in "${ALL_DBS[@]}"; do
  skip=
  for j in "${SYSTEM_DBS[@]}"; do
    [[ $i == $j ]] && { skip=1; break; }
  done
  [[ -n $skip ]] || DBS+=("$i")
done

for db in "${DBS[@]}"; do
  echo "Dumping: $db..."
  $MYSQLDUMP_PATH --opt --skip-add-locks -h $MYSQL_HOST -u$MYSQL_USER -p$MYSQL_PASS $db | gzip > $MYSQL_BACKUP_DIR$db\_$THE_DATE
done

# Delete old dumps
echo "-----------------------------------"
echo "Deleting old backups..."
# List dumps to be deleted to stdout (for report)
$FIND_PATH $MYSQL_BACKUP_DIR*.sql.gz -mtime +$KEEP_MYSQL
# Delete dumps older than specified number of days
$FIND_PATH $MYSQL_BACKUP_DIR*.sql.gz -mtime +$KEEP_MYSQL -exec rm {} +


# Announce the completion time
echo "-----------------------------------"
echo "Backup Completed: $(date)"
```