DISASTER RECOVERY WITH IBM CLOUD VIRTUAL SERVERS

BATCH MEMBER

711221243032:NITHYASHREE.K.N

PHASE-3

TOPIC: FLOODS DISASTER RECOVERY WITH IBM CLOUD VIRTUAL SERVICES.

INTRODUCTION :

Introducing an innovative approach to world disaster risk management on IBM Cloud virtual servers, where data-driven insights, advanced technology, and global collaboration converge. This initiative aims to revolutionize disaster preparedness, early warning systems, and response strategies, bridging geographical boundaries for a more resilient and coordinated global disaster management system. By leveraging the power of cloud computing and IBM's expertise, we are redefining how the world addresses the increasing challenges of natural and man-made disasters.

Requirement :

- IBM Cloud Infrastructure
- Data Backup and Replication
- Disaster Recovery Plan
- Redundancy and Failover
- Monitoring and Alerting
- Security Measures
- Network Connectivity
- Testing and Simulation
- Documentation and Reporting
- Compliance and Regulations
- Community Engagement
- Continuous Improvement

Different Functions Based on Requirements :

1. Data Protection and Backup :
- Function: Safeguarding critical data by regularly backing it up to secure cloud storage, ensuring data integrity and availability.
- Importance: Protects against data loss and provides a foundation for recovery.

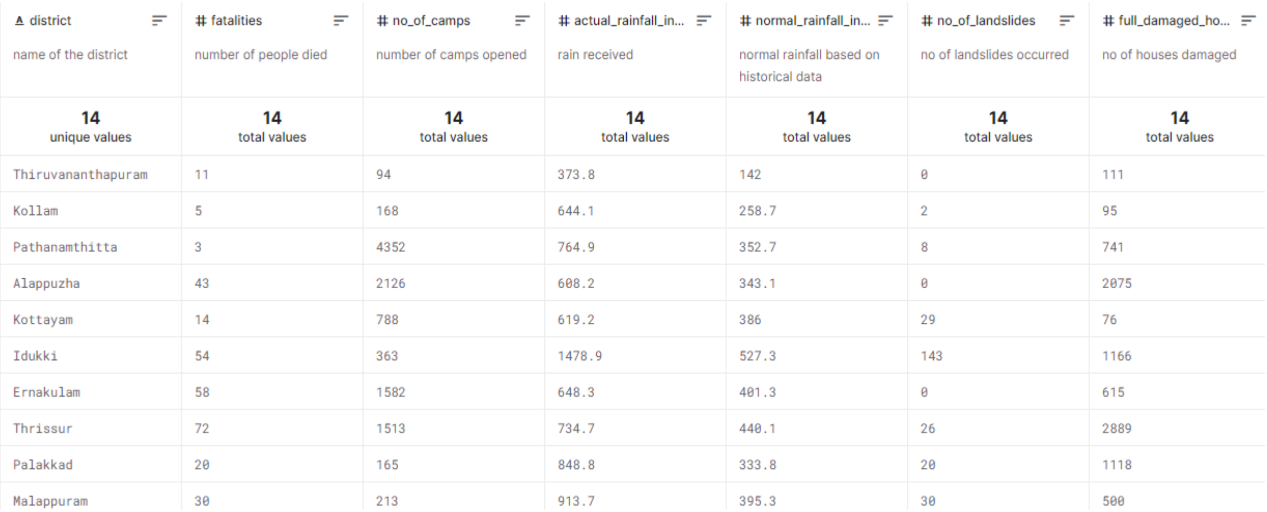
2. Data Replication and Redundancy:
- Function: Replicating data in real-time to a secondary location in the IBM Cloud, ensuring data redundancy and minimal downtime.
- Importance: Enables rapid recovery with minimal data loss.

3. Disaster Recovery Planning:
- Function: Developing a well-documented disaster recovery plan outlining procedures, roles, and communication strategies in a flood event.
- Importance: Ensures a structured and organized response to disasters.

4. Monitoring and Alerting:
- Function: Continuously monitoring the health and performance of IBM Cloud servers and alerting in case of issues.
- Importance: Provides early detection of problems and allows for prompt corrective actions.

5. Testing and Documentation:
- Function: Regularly testing disaster recovery procedures through simulations and maintaining comprehensive documentation.
- Importance: Validates the effectiveness of the recovery plan and ensures that it is up-to-date and well-understood by stakeholders.

Flood Dataset :



district	fatalities	no_of_camps	actual_rainfall_in...	normal_rainfall_in...	no_of_landslides	full_damaged_ho...
name of the district	number of people died	number of camps opened	rain received	normal rainfall based on historical data	no of landslides occurred	no of houses damaged
14 unique values	14 total values	14 total values	14 total values	14 total values	14 total values	14 total values
Thiruvananthapuram	11	94	373.8	142	0	111
Kollam	5	168	644.1	258.7	2	95
Pathanamthitta	3	4352	764.9	352.7	8	741
Alappuzha	43	2126	608.2	343.1	0	2075
Kottayam	14	788	619.2	386	29	76
Idukki	54	363	1478.9	527.3	143	1166
Ernakulam	58	1582	648.3	401.3	0	615
Thrissur	72	1513	734.7	440.1	26	2889
Palakkad	20	165	848.8	333.8	20	1118
Malappuram	30	213	913.7	395.3	30	500

Coding :

Import packages

```
library(tidyverse)
library(ggplot2)
library(viridis)
library(reshape2)
library(plotly)

theme_set(
  theme_minimal() +
    theme(legend.position = "right")
)
```

Importing dataset containing warning information

```
df = read_csv('../input/kerala-flood2018//warnings_actual_predicted.csv')
```

Viewing the data

```
colnames(df)

summary(df)

head(df)
```

Preprocessing

```
df$date = as.Date(df$date, format = '%m/%d/%Y')

df$actual_rainfall = factor(df$actual_rainfall,
                            levels = c('Green', 'Yellow', 'Orange',
'Red'),
                            ordered=TRUE)

df$predicted_rainfall = factor(df$predicted_rainfall,
                               levels = c('Green', 'Yellow', 'Orange',
'Red'),
                               ordered=TRUE)

df_temp <- df %>%
    filter(actual_rainfall != 'Green')


head(df)
```

Rain Intensity

```
options(repr.plot.width = 8, repr.plot.height = 7)
ggplot(data = df_temp, aes(x=actual_rainfall, fill=actual_rainfall))+
    geom_bar()+
    geom_text(stat='count', aes(label=..count..), position =
position_stack(vjust = 0.5))+
    scale_fill_manual(values = c("#DFDF00", "#FFA836", "#FC3107"))+
    guides(fill=guide_legend(title="Warning"))+
    labs(title='Number rains with different intensity', x="Rain
Intensity", y='No. warnings')+
    theme(text=element_text(size=16))
```

District wise rainfall

```
options(repr.plot.width = 15, repr.plot.height = 12)
ggplot(data = df_temp, aes(x=actual_rainfall, fill=actual_rainfall))+
    geom_bar()+
    scale_fill_manual(values = c("#DFDF00", "#FFA836", "#FC3107"))+
    facet_wrap(vars(date))+
    labs(title='Day wise rainfall')+
    theme(text=element_text(size=16))+
    theme(panel.spacing = unit(4, "lines"))
```

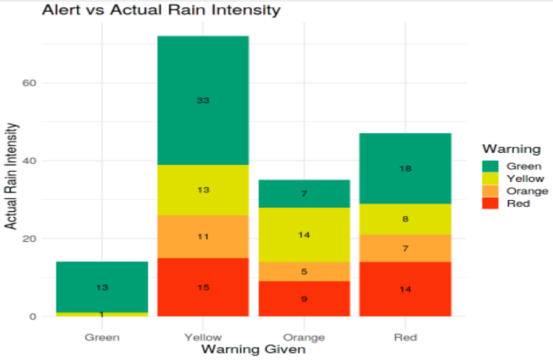
Getting the data before 20th August

```
df_main <- df %>%
    filter(date < '2018-08-20')

head(df_main)
```

Alert vs Actual Rain

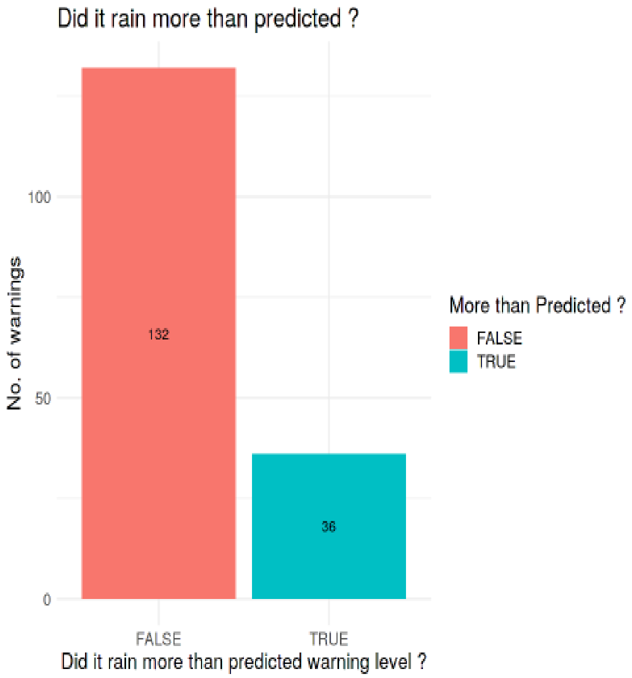
```
options(repr.plot.width = 8, repr.plot.height = 7)
ggplot(data = df_main, aes(x=predicted_rainfall,
fill=actual_rainfall))+
    geom_bar()+
    geom_text(stat='count', aes(label=..count..), position =
position_stack(vjust = 0.5))+
    scale_fill_manual(values = c("#009E73", "#DFDF00", "#FFA836",
"#FC3107"))+
    guides(fill=guide_legend(title="Warning"))+
    labs(title='Alert vs Actual Rain Intensity', x="Warning Given",
y='Actual Rain Intensity')+
    theme(text=element_text(size=16))
```

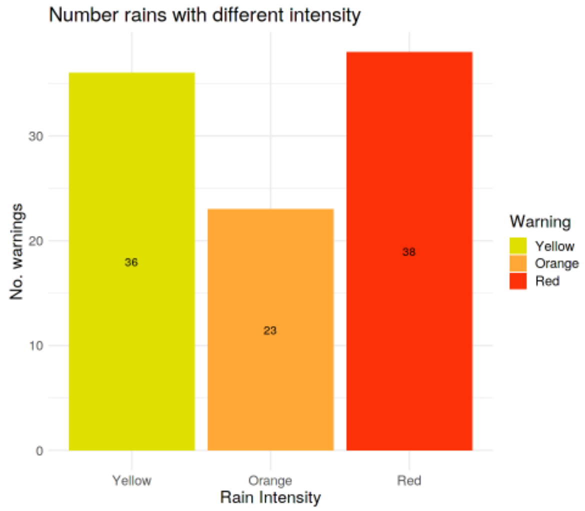


Did it rain more than predicted ?

```
df_main$more_than_pred <-
df_main$actual_rainfall>df_main$predicted_rainfall
head(df_main)

options(repr.plot.width = 8, repr.plot.height = 7)
ggplot(data = df_main, aes(more_than_pred, fill=more_than_pred))+
    geom_bar()+
    geom_text(stat='count', aes(label=..count..), position =
position_stack(vjust = 0.5))+
    guides(fill=guide_legend(title="More than Predicted ?"))+
    labs(title='Did it rain more than predicted ?', x="Did it rain more
than predicted warning level ?", y='No. of warnings')+
   theme(text=element_text(si
```





Conclusion

1. During the time of climate change, predicting the the weather has been tough challange for authorities across the world.
2. Although the the predictions and alert given were on spot and off in different places and dates in Kerala during 2018 flood time, it looks like IMD has done a descent job (Up for the debate).
3. But it has to be noted that there were times when it rained with intesity RED when there were only a YELLOW alert.