```
# OPMIG-09: Troubleshooting & Validation

> **Series:** OPMIG | **Notebook:** 9 of 9 | **Created:** December 2025
> **Level:** Intermediate
> **Prerequisites:** OPMIG-01 through OPMIG-08
> **Estimated Time:** 45 minutes

---

## Learning Objectives

By the end of this notebook, you will be able to:

- ⭐ **Navigate decision trees** to diagnose common OpenPipeline issues
- **Validate pipeline processing** using DQL verification queries
- **Troubleshoot parsing failures** and identify root causes
- **Debug masking and security processing** to ensure PII/PHI protection
- ⭐ **Execute emergency rollback procedures** when issues occur
- ⭐ **Resolve performance problems** in high-volume environments
- **Identify bucket and routing issues** with systematic diagnosis
- **Test end-to-end pipeline flows** before production deployment

---



---

## Migration Validation Checklist

Use this checklist to validate your migration:

### Data Flow Validation

- [ ] All log sources are flowing to OpenPipeline
- [ ] Data volumes match expectations
- [ ] No data loss detected
- [ ] Timestamps are correct

### Parsing Validation

- [ ] Log levels extracted correctly
- [ ] Custom fields parsed as expected
- [ ] JSON payloads properly flattened
- [ ] No parsing errors in logs

### Routing Validation
```

```
- [ ] Logs routed to correct pipelines
- [ ] Bucket assignments are correct
- [ ] No unrouted data accumulating

### Security Validation

- [ ] Sensitive data masked properly
- [ ] No PII visible in stored logs
- [ ] Compliance requirements met

### Extraction Validation

- [ ] Metrics being generated
- [ ] Events appearing correctly
- [ ] Business events available

---

## Troubleshooting Decision Trees

Visual decision trees for diagnosing and resolving common OpenPipeline
issues.

### Decision Tree 1: Logs Not Appearing

![Troubleshooting Decision Tree]
(
ciIHZpZXdCb3g9IjAgMCA4MDAgMzgwIj4KICA8ZGVmcz4KICAgIDxsaW5lYXJHcmFkaWVudCBpZD0
ic3RhcnRHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAlIj4KICAgICAgPHN0
b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6I2VmNDQ0NDtzdG9wLW9wYWNpdHk6MSIgL
z4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojZGMyNjI2O3N0b3
Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5lYXJHcmFkaWVudCB
pZD0icXVlc3Rpb25HcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAlIj4KICAg
ICAgPHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6IzNiODJmNjtzdG9wLW9wYWNpd
Hk6MSIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojMjU2M2
ViO3N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5lYXJHcmF
kaWVudCBpZD0iZml4R3JhZCIgeDE9IjAlIiB5MT0iMCUiIHgyPSIxMDAlIiB5Mj0iMTAwJSI+CiAg
ICAgIDxzdG9wIG9mZnNldD0iMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiMyMmM1NWU7c3RvcC1vcGFja
XR5OjEiIC8+CiAgICAgIDxzdG9wIG9mZnNldD0iMTAwJSIgc3R5bGU9InN0b3AtY29sb3I6IzE2YT
M0YTtzdG9wLW9wYWNpdHk6MSIgLz4KICAgIDwvbGluZWFyR3JhZGllbnQ+CiAgICA8ZmlsdGVyIGl
kPSJkdFNoYWRvdyI+CiAgICAgIDxmZURyb3BTaGFkb3cgZHg9IjIiIGR5PSIxIiBzdGREZXZpYXRp
b249IjIiIGZsb29kLW9wYWNpdHk9IjAuMTUiLz4KICAgIDwvZmlsdGVyPgogICAgPGZpbHRlciBpZ
D0iZHRBcnJvdyIgbWFya2VyV2lkdGg9IjEwIiBtYXJrZXJIZWlnaHQ9IjciIHJlZlg9IjciIHJlZl
k9IjMuNSIgb3JpZW50PSJhdXRvIj4KICAgICAgPHBvbHlnb24gcG9pbnRzPSIwIDAsIDEwIDMuNSw
gMCA3IiBmaWxsPSIjNjQ3NDhiIi8+CiAgICA8L21hcmtlcj4KICAgIDxtYXJrZXIgaWQ9Im5vQXJy
b3ciIG1hcmtlcldpZHRoPSIxMCIgbWFya2VySGVpZ2h0PSI3IiByZWZYPSI5IiByZWZZPSIzLjUiI
G9yaWVudD0iYXV0byI+CiAgICAgIDxwb2x5Z29uIHBvaW50cz0iMCAwLCAxMCAzLjUsIDAgNyIgZm
lsbD0iI2RjMjYyNiIvPgogICAgPC9tYXJrZXI+CiAgPC9kZWZzPgogICA8bWFya2VyIGlkPSJ5ZXNBcnJvdyIgbWF
```

ya2VyV2lkdGg9IjEwIiBtYXJrZXJIZWlnaHQ9IjciIHJlZlg9IjkiIHJlZlk9IjMuNSIgb3JpZW50
PSJhdXRvIj4KICAgICAgPHBvbHlnb24gcG9pbnRzPSIwIDAsIDEwIDMuNSwgMCA3IiBmaWxsPSIjM
TZhMzRhIi8+CiAgICA8L21hcmtlcj4KICA8L2RlZnM+CgogIDwhLS0gQmFja2dyb3VuZCAtLT4KIC
A8cmVjdCB3aWR0aD0iODAwIiBoZWlnaHQ9IjM4MCIgZmlsbD0iI2Y4ZjlmYSIgcng9IjEwIi8+Cgo
gIDwhLS0gVGl0bGUgLS0+CiAgPHRleHQgeD0iNDAwIiB5PSIyOCIgZm9udC1mYW1pbHk9IkFyaWFs
LCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjE4IiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0iIzMzM
yIgdGV4dC1hbmNob3I9Im1pZGRsZSI+VHJvdWJsZXNob290aW5nIERlY2lzaW9uIFRyZWU6IExvZ3
MgTm90IEFwcGVhcmluZzwvdGV4dD4KCiAgPCEtLSBTdGFydCAtLT4KICA8cmVjdCB4PSIzNDAiIHk
9IjUwIiB3aWR0aD0iMTIwIiBoZWlnaHQ9IjQwIiByeD0iMjAiIGZpbGw9InVybCgjc3RhcnRHcmFk
KSIgZmlsdGVyPSJ1cmwoI2R0U2hhZG93KSIvPgogIDx0ZXh0IHg9IjQwMCIgeT0iNzUiIGZvbnQtZ
mFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZm9udC13ZWlnaHQ9ImJvbG
QiIGZpbGw9IndoaXRlIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5Mb2dzIE1pc3NpbmcgPC90ZXh0Pgo
KICA8cGF0aCBkPSJNNDAwLDkwIEw0MDAsMTE1IiBzdHJva2U9IiM2NDc0OGIiIHN0cm9rZS13aWR0
aD0iMiIgZmlsbD0ibm9uZSIgbWFya2VyLWVuZD0idXJsKCNkdEFycm93KSIvPgoKICA8IS0tIFF1Z
XN0aW9uIDE6IEluZ2VzdGlvbiAtLT4KICA8cmVjdCB4PSIzMDUiIHk9IjEyMCIgd2lkdGg9IjE5MC
IgaGVpZ2h0PSI0NSIgcng9IjQiIGZpbGw9InVybCgjcXVlc3Rpb25HcmFkKSIgZmlsdGVyPSJ1cmw
oI2R0U2hhZG93KSIvPgogIDx0ZXh0IHg9IjQwMCIgeT0iMTQyIiBmb250LWZhbWlseT0iQXJpYWws
IHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSJ3aGl0Z
SIgdGV4dC1hbmNob3I9Im1pZGRsZSI+MS4gQXJlIGxvZ3MgYmVpbmcgaW5nZXN0ZWQ/PC90ZXh0Pg
ogIDx0ZXh0IHg9IjQwMCIgeT0iMTU4IiBmb250LWZhbWlseT0ibW9ub3NwYWNlIiBmb250LXNpemU
9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiIHRleHQtYW5jaG9yPSJtaWRkbGUiPmZl
dGNoIGxvZ3MgfCBzdW1tYXJpemUgY291bnQoKTwvdGV4dD4KCiAgPCEtLSBObyBicmFuY2ggdG8gZ
ml4IC0tPgogIDxwYXRoIGQ9Ik0zMDUsMTQyIEwyMjAsMTQyIiBzdHJva2U9IiNkYzI2MjYiIHN0cm
9rZS13aWR0aD0iMiIgZmlsbD0ibm9uZSIgbWFya2VyLWVuZD0idXJsKCNub0Fycm93KSIvPgogIDx
0ZXh0IHg9IjI2MiIgeT0iMTM1IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQt
c2l6ZT0iMTAiIGZpbGw9IiNkYzI2MjYiIHRleHQtYW5jaG9yPSJtaWRkbGUiPk5vPC90ZXh0PgoKI
CA8cmVjdCB4PSI1MCIgeT0iMTIwIiB3aWR0aD0iMTYwIiBoZWlnaHQ9IjQ1IiByeD0iNCIgZmlsbD
0iI2ZlY2FjYSIgc3Ryb2tlPSIjZGMyNjI2IiBzdHJva2Utd2lkdGg9IjEiLz4KICA8dGV4dCB4PSI
xMzAiIHk9IjEzOCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmb250LXdlaWdodD0iYm9sZCIg
ZmlsbD0iI2RjMjYyNiIgdGV4dC1hbmNob3I9Im1pZGRsZSI+Q2hlY2sgSW5nZXN0aW9uPC90ZXh0P
gogIDx0ZXh0IHg9IjEzMCIgeT0iMTUyIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIG
ZvbnQtc2l6ZT0iMTAiIGZpbGw9IiM3ZjFkMWQiIHRleHQtYW5jaG9yPSJtaWRkbGUiPk9uZUFnZW5
0LCBBUEkgdG9rZW4sIGVuZHBvaW50PC90ZXh0PgoKICA8IS0tIFllcyB0byBuZXh0IC0tPgogIDxw
YXRoIGQ9Ik00MDAsMTY1IEw0MDAsMTk1IiBzdHJva2U9IiMxNmEzNGEiIHN0cm9rZS13aWR0aD0iM
iIgZmlsbD0ibm9uZSIgbWFya2VyLWVuZD0idXJsKCN5ZXNBcnJvdykiLz4KICA8dGV4dCB4PSI0MT
AiIHk9IjE4MiIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiB
maWxsPSIjMTZhMzRhIj5ZZXM8L3RleHQ+CgogIDwhLS0gUXVlc3Rpb24gMjogRHJvcCAtLT4KICA8
cmVjdCB4PSIzMDUiIHk9IjIwMCIgd2lkdGg9IjE5MCIgaGVpZ2h0PSI0NSIgcng9IjQiIGZpbGw9I
nVybCgjcXVlc3Rpb25HcmFkKSIgZmlsdGVyPSJ1cmwoI2R0U2hhZG93KSIvPgogIDx0ZXh0IHg9Ij
QwMCIgeT0iMjIyIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTA
iIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+Mi4g
QmVpbmcgZHJvcHBlZD88L3RleHQ+CiAgPHRleHQgeD0iNDAwIiB5PSIyMzgiIGZvbnQtZmFtaWx5P
SJtb25vc3BhY2UiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSIgdG
V4dC1hbmNob3I9Im1pZGRsZSI+Q2hlY2sgZHJvcCBwcm9jZXNzb3IgY29uZGl0aW9uczwvdGV4dD4
KCiAgPCEtLSBZZXMgYnJhbmNoIHRvIGZpeCAtLT4KICA8cGF0aCBkPSJNNDk1LDIyMiBMNTgwLDIy
MiIgc3Ryb2tlPSIjZGMyNjI2IiBzdHJva2Utd2lkdGg9IjIiIGZpbGw9Im5vbmUiIG1hcmtlci1lb
mQ9InVybCgjbm9BcnJvdykiLz4KICA8dGV4dCB4PSI1MzgiIHk9IjIxNSIgZm9udC1mYW1pbHk9Ik
FyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjZGMyNjI2IiB0ZXh0LWFuY2h
vcj0ibWlkZGxlIj5ZZXM8L3RleHQ+CiAgPHJlY3QgeD0iNTg1IiB5PSIyMDAiIHdpZHRoPSIxNjAi
IGhlaWdodD0iNDUiIHJ4PSI0IiBmaWxsPSIjZmVjYWNhIiBzdHJva2U9IiNkYzI2MjYiIHN0cm9rZ
S13aWR0aD0iMSIvPgogIDx0ZXh0IHg9IjY2NSIgeT0iMjE4IiBmb250LWZhbWlseT0iQXJpYWwsIH
NhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjZGMyNjI
2Ij5ZZXM8L3RleHQ+CiAgPHRleHQgeD0iNDAwIiB5PSIyMzgiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iI2RjMjYyNiIgZmlsbD0iI2RjMjYyNiIgZmlsbD0iI2RjMjYyNiIgZmlsbD0iI2RjMjYyNiIgZmlsbD0iI2RjMjYyNiIgZmlsbD0iI2RjMjYyNiIg

```
+CgogIDxyZWN0IHg9IjUwIiB5PSIyMDAiIHdpZHRoPSIxNjAiIGhlaWdodD0iNDUiIHJ4PSI0IiBm
aWxsPSIjZmVmM2M3IiBzdHJva2U9IiNmNTllMGIiIHN0cm9rZS13aWR0aD0iMSIvPgogIDx0ZXh0I
Hg9IjEzMCIgeT0iMjE4IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT
0iMTAiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjOTI0MDBlIiB0ZXh0LWFuY2hvcj0ibWlkZGx
lIj5BZGp1c3QgRHJvcCBSdWxlczwvdGV4dD4KICA8dGV4dCB4PSIxMzAiIHk9IjIzMiIgZm9udC1m
YW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjNzgzNTBmIiB0Z
Xh0LWFuY2hvcj0ibWlkZGxlIj5NYWtlIGNvbmRpdGlvbnMgbW9yZSBzcGVjaWZpYzwvdGV4dD4KCi
AgPCEtLSBObyB0byBuZXh0IC0tPgogIDxwYXRoIGQ9Ik00MDAsMjQ1IEw0MDAsMjc1IiBzdHJva2U
9IiMxNmEzNGEiIHN0cm9rZS13aWR0aD0iMiIgZmlsbD0ibm9uZSIgbWFya2VyLWVuZD0idXJsKCN5
ZXNBcnJvdykiLz4KICA8dGV4dCB4PSI0MTAiIHk9IjI2MiIgZm9udC1mYW1pbHk9IkFyaWFsLCBzY
W5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjMTZhMzRhIj5ObzwvdGV4dD4KCiAgPCEtLS
BRdWVzdGlvbiAzOiBSb3V0aW5nIC0tPgogIDxyZWN0IHg9IjMwNSIgeT0iMjgwIiB3aWR0aD0iMTk
wIiBoZWlnaHQ9IjQ1IiByeD0iNCIgZmlsbD0idXJsKCNxdWVzdGlvbkdyYWQpIiBmaWx0ZXI9InVy
bCgjZHRTaGFkb3cpIi8+CiAgPHRleHQgeD0iNDAwIiB5PSIzMDIiIGZvbnQtZmFtaWx5PSJBcmlhb
Cwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IndoaX
RlIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj4zLiBSb3V0ZWQgdG8gcmlnaHQgcGlwZWxpbmU/PC90ZXh
0PgogIDx0ZXh0IHg9IjQwMCIgeT0iMzE4IiBmb250LWZhbWlseT0ibW9ub3NwYWNlIiBmb250LXNp
emU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiIHRleHQtYW5jaG9yPSJtaWRkbGUiP
kNoZWNrIGR0Lm9wZW5waXBlbGluZS5waXBlbGluZXM8L3RleHQ+CgogIDwhLS0gTm8gYnJhbmNoIH
RvIGZpeCAtLT4KICA8cGF0aCBkPSJNMzA1LDMwMiBMMjIwLDMwMiIgc3Ryb2tlPSIjZGMyNjI2IiB
zdHJva2Utd2lkdGg9IjIiIGZpbGw9Im5vbmUiIG1hcmtlci1lbmQ9InVybCgjbm9BcnJvdykiLz4K
ICA8dGV4dCB4PSIyNjIiIHk9IjI5NSIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb
250LXNpemU9IjEwIiBmaWxsPSIjZGMyNjI2Ij5ObzwvdGV4dD4KCiAgPHJlY3QgeD0iNTAiIHk9Ij
I4MCIgd2lkdGg9IjE2MCIgaGVpZ2h0PSI0NSIgcng9IjQiIGZpbGw9IiNmZWYyZjciIHN0cm9rZT0
iI2Y1OWUwYiIgc3Ryb2tlLXdpZHRoPSIxIi8+CiAgPHRleHQgeD0iMTMwIiB5PSIyOTgiIGZvbnQt
ZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZm9udC13ZWlnaHQ9ImJvb
GQiIGZpbGw9IiM5MjQwMGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPkZpeCBSb3V0ZSBDb25kaXRpb2
5zPC90ZXh0PgogIDx0ZXh0IHg9IjEzMCIgeT0iMzEyIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnM
tc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiM3ODM1MGYiIHRleHQtYW5jaG9yPSJtaWRkbGUi
PlVwZGF0ZSBtYXRjaGluZyBjb25kaXRpb248L3RleHQ+CgogIDwhLS0gWWVzIHRvIG5leHQgcXVlc
3Rpb24gLS0+CiAgPHBhdGggZD0iTTQ5NSwzMDIgTDU1MCwzMDIiIHN0cm9rZT0iIzE2YTM0YSIgc3
Ryb2tlLXdpZHRoPSIyIiBmaWxsPSJub25lIiBtYXJrZXItZW5kPSJ1cmwoI3llc0Fycm93KSIvPgo
gIDx0ZXh0IHg9IjUyMiIgeT0iMjk1IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZv
bnQtc2l6ZT0iMTAiIGZpbGw9IiMxNmEzNGEiPllllczwvdGV4dD4KCiAgPCEtLSBRdWVzdGlvbiA0O
iBCdWNrZXQgLS0+CiAgPHJlY3QgeD0iNTYwIiB5PSIyODAiIHdpZHRoPSIxOTAiIGhlaWdodD0iND
UiIHJ4PSI0IiBmaWxsPSJ1cmwoI3F1ZXN0aW9uR3JhZCkiIGZpbHRlcj0idXJsKCNkdFNoYWRvdyk
iLz4KICA8dGV4dCB4PSI2NTUiIHk9IjMwMiIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlm
IiBmb250LXNpemU9IjEwIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0id2hpdGUiIHRleHQtYW5ja
G9yPSJtaWRkbGUiPjQuIEluIGNvcnJlY3QgYnVja2V0PzwvdGV4dD4KICA8dGV4dCB4PSI2NTUiIH
k9IjMxOCIgZm9udC1mYW1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSg
yNTUsMjU1LDI1NSwwLjkpIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5DaGVjayBkdC5zeXN0ZW0uYnVj
a2V0PC90ZXh0PgoKICA8IS0tIE5vIGJyYW5jaCB0byBmaXggLS0+CiAgPHBhdGggZD0iTTY1NSwzM
jUgTDY1NSwzNTAiIHN0cm9rZT0iI2RjMjYyNiIgc3Ryb2tlLXdpZHRoPSIyIiBmaWxsPSJub25lIi
BtYXJrZXItZW5kPSJ1cmwoI25vQXJyb3cpIi8+CiAgPHRleHQgeD0iNjY1IiB5PSIzNDAiIGZvbnQ
tZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iI2RjMjYyNiI+
Tm88L3RleHQ+CgogIDxyZWN0IHg9IjU2MCIgeT0iMzUwIiB3aWR0aD0iMTkwIiBoZWlnaHQ9IjI1I
iByeD0iNCIgZmlsbD0iI2ZlZjNjNyIgc3Ryb2tlPSIjZjU5ZTBiIiBzdHJva2Utd2lkdGg9IjEiLz
4KICA8dGV4dCB4PSI2NTUiIHk9IjM2NyIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiB
```

mb250LXNpemU9IjEwIiBmaWxsPSIjOTI0MDBlIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5GaXggcGlw
ZWxpbmUgc3RvcmFnZSBzZXR0aW5nczwvdGV4dD4KCiAgPCEtLSBZZXMgdG8gc3VjY2VzcyAtLT4KI
CA8cGF0aCBkPSJNNzUwLDMwMiBMNzcwLDMwMiBMNzcwLDIzMCBMNzU1LDIzMCIgc3Ryb2tlPSIjMT
ZhMzRhIiBzdHJva2Utd2lkdGg9IjIiIGZpbGw9Im5vbmUiIG1hcmtlci1lbmQ9InVybCgjeWVzQXJ
yb3cpIi8+CiAgPHRleHQgeD0iNzYwIiB5PSIyNjAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1z
ZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzE2YTM0YSIgdHJhbnNmb3JtPSJyb3RhdGUoLTkwL
CA3NjAsIDI2MCkiPllllczwvdGV4dD4KCiAgPCEtLSBTdWNjZXNzIC0tPgogIDxyZWN0IHg9IjYxMC
IgeT0iMjAwIiB3aWR0aD0iMTQwIiBoZWlnaHQ9IjQ1IiByeD0iNCIgZmlsbD0idXJsKCNmaXhHcmF
kKSIgZmlsdGVyPSJ1cmwoI2R0U2hhZG93KSIvPgogIDx0ZXh0IHg9IjY4MCIgeT0iMjIyIiBmb250
LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZvbnQtd2VpZ2h0PSJib
2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+4pyFIFN1Y2Nlc3MhPC90ZXh0Pg
ogIDx0ZXh0IHg9IjY4MCIgeT0iMjM4IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZ
vbnQtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1hbmNob3I9Im1p
ZGRsZSI+TG9ncyB2aXNpYmxlIGluIHF1ZXJpZXM8L3RleHQ+CgogIDwhLS0gQWRkaXRpb25hbCBwY
XRocyBmcm9tIGZpeGVzIHRvIGluZ2VzdGlvbi9zdWNjZXNzIC0tPgogIDxwYXRoIGQ9Ik0xMzAsMT
Y1IEwxMzAsMTkwIiBzdHJva2U9IiMxNmEzNGEiIHN0cm9rZS13aWR0aD0iMSIgc3Ryb2tlLWRhc2h
hcnJheT0iNCwyIiBmaWxsPSJub25lIi8+CiAgPHRleHQgeD0iMTQwIiB5PSIxNzgiIGZvbnQtZmFt
aWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzE2YTM0YSI+Zml4Z
WQ/PC90ZXh0Pgo8L3N2Zz4K)

Decision Tree 2: Parsing Failures

![Parsing Failures Decision Tree]
(
ciIHZpZXdCb3g9IjAgMCA4MDAgNjUwIj4KICA8ZGVmcz4KICAgIDxsaW5lYXJHcmFkaWVudCBpZD0
icHJvYmxlbUdyYWQiIHgxPSIwJSIgeTE9IjAlIiB4Mj0iMTAwJSIgeTI9IjEwMCUiPgogICAgICA8
c3RvcCBvZmZzZXQ9IjAlIiBzdHlsZT0ic3RvcC1jb2xvcjojZWY0NDQ0O3N0b3Atb3BhY2l0eToxI
iAvPgogICAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiNkYzI2MjY7c3
RvcC1vcGFjaXR5OjEiIC8+CiAgICA8L2xpbmVhckdyYWRpZW50PgogICAgPGxpbmVhckdyYWRpZW5
0IGlkPSJzdGVwR3JhZCIgeDE9IjAlIiB5MT0iMCUiIHgyPSIxMDAlIiB5Mj0iMTAwJSI+CiAgICAg
IDxzdG9wIG9mZnNldD0iMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiMzYjgyZjY7c3RvcC1vcGFjaXR5O
jEiIC8+CiAgICAgIDxzdG9wIG9mZnNldD0iMTAwJSIgc3R5bGU9InN0b3AtY29sb3I6IzI1NjNlYj
tzdG9wLW9wYWNpdHk6MSIgLz4KICAgIDwvbGluZWFyR3JhZGllbnQ+CiAgICA8bGluZWFyR3JhZGl
lbnQgaWQ9Inllc0dyYWQiIHgxPSIwJSIgeTE9IjAlIiB4Mj0iMTAwJSIgeTI9IjEwMCUiPgogICAg
ICA8c3RvcCBvZmZzZXQ9IjAlIiBzdHlsZT0ic3RvcC1jb2xvcjojMTBiOTgxO3N0b3Atb3BhY2l0e
ToxIiAvPgogICAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiMwNTk2Nj
k7c3RvcC1vcGFjaXR5OjEiIC8+CiAgICA8L2xpbmVhckdyYWRpZW50PgogICAgPGxpbmVhckdyYWR
pZW50IGlkPSJub0dyYWQiIHgxPSIwJSIgeTE9IjAlIiB4Mj0iMTAwJSIgeTI9IjEwMCUiPgogICAg
ICA8c3RvcCBvZmZzZXQ9IjAlIiBzdHlsZT0ic3RvcC1jb2xvcjojZjU5ZTBiO3N0b3Atb3BhY2l0e
ToxIiAvPgogICAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiNkOTc3MD
Y7c3RvcC1vcGFjaXR5OjEiIC8+CiAgICA8L2xpbmVhckdyYWRpZW50PgogICAgPGxpbmVhckdyYWR
pZW50IGlkPSJmaXhHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAlIj4KICAg
ICAgPHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6IzhiNWNmNjtzdG9wLW9wYWNpd
Hk6MSIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojN2MzYW

```
VkO3N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxmaWx0ZXIgaWQ
9ImR0U2hhZG93Ij4KICAgICAgPGZlRHJvcFNoYWRvdyBkeD0iMiIgZHk9IjIiIHN0ZERldmlhdGlv
bj0iMyIgZmxvb2Qtb3BhY2l0eT0iMC4xNSIvPgogICAgPC9maWx0ZXI+CiAgICA8bWFya2VyIGlkP
SJkdEFycm93IiBtYXJrZXJXaWR0aD0iMTAiIG1hcmtlckhlaWdodD0iNyIgcmVmWD0iOSIgcmVmWT
0iMy41IiBvcmllbnQ9ImF1dG8iPgogICAgICA8cG9seWdvbiBwb2ludHM9IjAgMCwgMTAgMy41LCA
wIDciIGZpbGw9IiM2NDc0OGIiLz4KICAgIDwvbWFya2VyPgogICAgPG1hcmtlciBpZD0ieWVzQXJy
b3ciIG1hcmtlcldpZHRoPSIxMCIgbWFya2VySGVpZ2h0PSI3IiByZWZPYPSI5IiByZWZZPSIzLjUiI
G9yaWVudD0iYXV0byI+CiAgICAgIDxwb2x5Z29uIHBvaW50cz0iMCAwLCAxMCAzLjUsIDAgNyIgZm
lsbD0iIzEwYjk4MSIvPgogICAgPC9tYXJrZXI+CiAgICA8bWFya2VyIGlkPSJub0Fycm93IiBtYXJ
rZXJXaWR0aD0iMTAiIG1hcmtlckhlaWdodD0iNyIgcmVmWD0iOSIgcmVmWT0iMy41IiBvcmllbnQ9
ImF1dG8iPgogICAgICA8cG9seWdvbiBwb2ludHM9IjAgMCwgMTAgMy41LCAwIDciIGZpbGw9IiNmN
TllMGIiLz4KICAgIDwvbWFya2VyPgogICAgPG1hcmtlciBpZD0ieWVsZGVmcz4KCiAgPCEtLSBCYWNrZ3JvdW5kIC0tPgogID
xyZWN0IHdpZHRoPSI4MDAiIGhlaWdodD0iNjUwIiBmaWxsPSIjMGYxNzJhIiByeD0iMTAiLz4KICA
gPCEtLSBUaXRsZSAtLT4KICA8dGV4dCB4PSI0MDAiIHk9IjM1IiBmb250LWZhbWlseT0ic3lzdGVt
LXVpLCAtYXBwbGUtc3lzdGVtLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjIwIiBmb250LXdlaWdod
D0iYm9sZCIgZmlsbD0iI2YxZjVmOSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+RGVjaXNpb24gVHJlZS
AyOiBQYXJzaW5nIEZhaWx1cmVzPC90ZXh0PgoKICA8IS0tIFByb2JsZW0gQm94IC0tPgogIDxyZWN
0IHg9IjIxZjVmOSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+RGVjaXNpb24gVHJlZS
AyOiBQYXJzaW5nIEZhaWx1cmVzPC90ZXh0PgoKICA8IS0tIFByb2JsZW0gQm94IC0tPgogIDxyZWN
0IHg9IjQwIiB5PSIzMDAiIHdpZHRoPSIzMDAiIGhlaWdodD0iNDUiIHJ4PSI4IiBmaWxsPSJ1cmwo
I3Byb2JsZW1HcmFkKSIgZmlsdGVyPSJ1cmwoI2R0U2hhZG93KSIvPgogIDx0ZXh0IHg9IjQwMCIge
T0iODMiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTQiIG
ZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5GaWVsZHM
gbm90IGV4dHJhY3RlZCBmcm9tIGxvZyBjb250ZW50PC90ZXh0PgoKICA8IS0tIEFycm93IFN0
ZXAgMSAtLT4KICA8bGluZSB4MT0iNDAwIiB5MT0iMTAwIiB4Mj0iNDAwIiB5Mj0iMTI1IiBzdHJva
2U9IiM2NDc0OGIiIHN0cm9rZS13aWR0aD0iMiIgbWFya2VyLWVuZD0idXJsKCNkdEFycm93KSIvPg
oKICA8IS0tIFN0ZXAgMSAtLT4KICA8cmVjdCB4PSIyMDAiIHk9IjEzMCIgd2lkdGg9IjQwMCIgaGV
pZ2h0PSI1MCIgcng9IiIgbWFya2VyLWVuZD0idXJsKCNkdEFycm93KSIvPg
oKICA8IS0tIFN0ZXAgMSAtLT4KICA8cmVjdCB4PSIyMDAiIHk9IjEzMCIgd2lkdGg9IjQwMCIgaGV
pZ2h0PSI1MCIgcng9IiIgbWFya2VyLWVuZD0idXJsKCNkdEFycm93KSIvPgoKICA8IS0tIFN0ZXAg
b3cpIi8+CiAgPGNpcmNsZSBjeD0iMjMwIiBjeT0iMTU1IiByPSIxNSIgZmlsbD0icmdiYSgyNTUsM
jU1LDI1NSwwLjIpIi8+CiAgPHRleHQgeD0iMjMwIiB5PSIxNjEiIGZvbnQtZmFtaWx5PSJzeXN0ZW
0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTQiIG
ZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj4xPC90ZXh0PgogIDx0ZXh0IHg9IjMwIiB5PSIyMDAiIHdpZHRoPSIzMDAiIGhlaWdodD0iNDUiIHJ4PSI4IiBmaWxsPSJ1cmwoI3Byb2JsZW1HcmFkKSIgZmlsdGVyPSJ1cmwoI2R0U2hhZG93KSIvPgoKICA8IS0tIFByb2JsZW0gQm94IC0tPgogIDxyZWN0IHg9IjIxZjVmOSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+RGVjaXNpb24gVHJlZS
AyOiBQYXJzaW5nIEZhaWx1cmVzPC90ZXh0PgoKICA8IS0tIFByb2JsZW0gQm94IC0tPgogIDxyZWN
0IHg9IjQwIiB5PSIzMDAiIHdpZHRoPSIzMDAiIGhlaWdodD0iNDUiIHJ4PSI4IiBmaWxsPSJ1cmwo
I3Byb2JsZW1HcmFkKSIgZmlsdGVyPSJ1cmwoI2R0U2hhZG93KSIvPgogIDx0ZXh0IHg9IjQwMCIge
T0iODMiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTQiIG
ZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj4yPC90ZXh0PgoKICA8IS0tIFN0ZXAgMSAtLT4KICA8cmVjdCB4PSIyMDAiIHk9IjEzMCIgd2lkdGg9IjQwMCIgaGV
pZ2h0PSI1MCIgcng9IiIgbWFya2VyLWVuZD0idXJsKCNkdEFycm93KSIvPg
oKICA8IS0tIFN0ZXAgMSAtLT4KICA8cmVjdCB4PSIyMDAiIHk9IjEzMCIgd2lkdGg9IjQwMCIgaGV
pZ2h0PSI1MCIgcng9IiIgbWFya2VyLWVuZD0idXJsKCNkdEFycm93KSIvPgoKICA8IS0tIFN0ZXAg
b3cpIi8+CiAgPGNpcmNsZSBjeD0iMjMwIiBjeT0iMTU1IiByPSIxNSIgZmlsbD0icmdiYSgyNTUsM
jU1LDI1NSwwLjIpIi8+CiAgPHRleHQgeD0iMjMwIiB5PSIxNjEiIGZvbnQtZmFtaWx5PSJzeXN0ZW
0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTQiIG
ZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj4xPC90ZXh0PgogIDx0ZXh0IHg9IjMwIiB5PSIyMDAiIHdpZHRoPSIzMDAiIGhlaWdodD0iNDUiIHJ4PSI4IiBmaWxsPSJ1cmwoI3Byb2JsZW1HcmFkKSIgZmlsdGVyPSJ1cmwoI2R0U2hhZG93KSIvPgoKICA8IS0tIFByb2JsZW0gQm94IC0tPgogIDxyZWN0IHg9IjIxZjVmOSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+RGVjaXNpb24gVHJlZS
AyOiBQYXJzaW5nIEZhaWx1cmVzPC90ZXh0PgoKICA8IS0tIFByb2JsZW0gQm94IC0tPgogIDxyZWN
0IHg9IjQwIiB5PSIzMDAiIHdpZHRoPSIzMDAiIGhlaWdodD0iNDUiIHJ4PSI4IiBmaWxsPSJ1cmwo
I3Byb2JsZW1HcmFkKSIgZmlsdGVyPSJ1cmwoI2R0U2hhZG93KSIvPgogIDx0ZXh0IHg9IjQwMCIge
T0iODMiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTQiIG
ZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj4yPC90ZXh0PgoKICA8IS0tIFN0ZXAgMSAtLT4KICA8cmVjdCB4PSIyMDAiIHk9IjEzMCIgd2lkdGg9IjQwMCIgaGV
pZ2h0PSI1MCIgcng9IiIgbWFya2VyLWVuZD0idXJsKCNkdEFycm93KSIvPg
oKICA8IS0tIFN0ZXAgMSAtLT4KICA8cmVjdCB4PSIyMDAiIHk9IjEzMCIgd2lkdGg9IjQwMCIgaGV
pZ2h0PSI1MCIgcng9IiIgbWFya2VyLWVuZD0idXJsKCNkdEFycm93KSIvPgoKICA8IS0tIFN0ZXAg
b3cpIi8+CiAgPGNpcmNsZSBjeD0iMjMwIiBjeT0iMTU1IiByPSIxNSIgZmlsbD0icmdiYSgyNTUsM
jU1LDI1NSwwLjIpIi8+CiAgPHRleHQgeD0iMjMwIiB5PSIxNjEiIGZvbnQtZmFtaWx5PSJzeXN0ZW
0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTQiIG
ZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj4zPC90ZXh0PgogIDx0ZXh0IHg9IjMwIiB5PSIyMDAiIHdpZHRoPSIzMDAiIGhlaWdodD0iNDUiIHJ4PSI4IiBmaWxsPSJ1cmwoI3Byb2JsZW1HcmFkKSIgZmlsdGVyPSJ1cmwoI2R0U2hhZG93KSIvPgoKICA8IS0tIFByb2JsZW0gQm94IC0tPgogIDxyZWN0IHg9IjIxZjVmOSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+RGVjaXNpb24gVHJlZS
AyOiBQYXJzaW5nIEZhaWx1cmVzPC90ZXh0PgoKICA8IS0tIFByb2JsZW0gQm94IC0tPgogIDxyZWN
0IHg9IjQwIiB5PSIzMDAiIHdpZHRoPSIzMDAiIGhlaWdodD0iNDUiIHJ4PSI4IiBmaWxsPSJ1cmwo
I3Byb2JsZW1HcmFkKSIgZmlsdGVyPSJ1cmwoI2R0U2hhZG93KSIvPgogIDx0ZXh0IHg9IjQwMCIge
T0iODMiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTQiIG
ZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj40PC90ZXh0PgoKICA8IS0tIFN0ZXAgMSAtLT4KICA8cmVjdCB4PSIyMDAiIHk9IjEzMCIgd2lkdGg9IjQwMCIgaGV
pZ2h0PSI1MCIgcng9IiIgbWFya2VyLWVuZD0idXJsKCNkdEFycm93KSIvPg
oKICA8IS0tIFN0ZXAgMSAtLT4KICA8cmVjdCB4PSIyMDAiIHk9IjEzMCIgd2lkdGg9IjQwMCIgaGV
pZ2h0PSI1MCIgcng9IiIgbWFya2VyLWVuZD0idXJsKCNkdEFycm93KSIvPgoKICA8IS0tIFN0ZXAg
b3cpIi8+CiAgPGNpcmNsZSBjeD0iMjMwIiBjeT0iMTU1IiByPSIxNSIgZmlsbD0icmdiYSgyNTUsM
jU1LDI1NSwwLjIpIi8+CiAgPHRleHQgeD0iMjMwIiB5PSIxNjEiIGZvbnQtZmFtaWx5PSJzeXN0ZW
0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTQiIG
ZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj41PC90ZXh0PgogIDx0ZXh0IHg9IjMwIiB5PSIyMDAiIHdpZHRoPSIzMDAiIGhlaWdodD0iNDUiIHJ4PSI4IiBmaWxsPSJ1cmwoI3Byb2JsZW1HcmFkKSIgZmlsdGVyPSJ1cmwoI2R0U2hhZG93KSIvPgoKICA8IS0tIFByb2JsZW0gQm94IC0tPgogIDxyZWN0IHg9IjIxZjVmOSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+RGVjaXNpb24gVHJlZS
AyOiBQYXJzaW5nIEZhaWx1cmVzPC90ZXh0PgoKICA8IS0tIFByb2JsZW0gQm94IC0tPgogIDxyZWN
0IHg9IjQwIiB5PSIzMDAiIHdpZHRoPSIzMDAiIGhlaWdodD0iNDUiIHJ4PSI4IiBmaWxsPSJ1cmwo
I3Byb2JsZW1HcmFkKSIgZmlsdGVyPSJ1cmwoI2R0U2hhZG93KSIvPgogIDx0ZXh0IHg9IjQwMCIge
T0iODMiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTQiIG
ZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj42PC90ZXh0PgoKICA8IS0tIFN0ZXAgMSAtLT4KICA8cmVjdCB4PSIyMDAiIHk9IjEzMCIgd2lkdGg9IjQwMCIgaGV
pZ2h0PSI1MCIgcng9IiIgbWFya2VyLWVuZD0idXJsKCNkdEFycm93KSIvPg
oKICA8IS0tIFN0ZXAgMSAtLT4KICA8cmVjdCB4PSIyMDAiIHk9IjEzMCIgd2lkdGg9IjQwMCIgaGV
pZ2h0PSI1MCIgcng9IiIgbWFya2VyLWVuZD0idXJsKCNkdEFycm93KSIvPgoKICA8IS0tIFN0ZXAg
b3cpIi8+CiAgPGNpcmNsZSBjeD0iMjMwIiBjeT0iMTU1IiByPSIxNSIgZmlsbD0icmdiYSgyNTUsM
jU1LDI1NSwwLjIpIi8+CiAgPHRleHQgeD0iMjMwIiB5PSIxNjEiIGZvbnQtZmFtaWx5PSJzeXN0ZW
0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTQiIG
ZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj43PC90ZXh0PgogIDx0ZXh0IHg9IjMwIiB5PSIyMDAiIHdpZHRoPSIzMDAiIGhlaWdodD0iNDUiIHJ4PSI4IiBmaWxsPSJ1cmwoI3Byb2JsZW1HcmFkKSIgZmlsdGVyPSJ1cmwoI2R0U2hhZG93KSIvPgoKICA8IS0tIFByb2JsZW0gQm94IC0tPgogIDxyZWN0IHg9IjIxZjVmOSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+RGVjaXNpb24gVHJlZS
AyOiBQYXJzaW5nIEZhaWx1cmVzPC90ZXh0PgoKICA8IS0tIFByb2JsZW0gQm94IC0tPgogIDxyZWN
0IHg9IjQwIiB5PSIzMDAiIHdpZHRoPSIzMDAiIGhlaWdodD0iNDUiIHJ4PSI4IiBmaWxsPSJ1cmwo
I3Byb2JsZW1HcmFkKSIgZmlsdGVyPSJ1cmwoI2R0U2hhZG93KSIvPgogIDx0ZXh0IHg9IjQwMCIge
T0iODMiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTQiIG
ZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj44PC90ZXh0PgoKICA8IS0tIFN0ZXAgMSAtLT4KICA8cmVjdCB4PSIyMDAiIHk9IjEzMCIgd2lkdGg9IjQwMCIgaGV
pZ2h0PSI1MCIgcng9IiIgbWFya2VyLWVuZD0idXJsKCNkdEFycm93KSIvPg
oKICA8IS0tIFN0ZXAgMSAtLT4KICA8cmVjdCB4PSIyMDAiIHk9IjEzMCIgd2lkdGg9IjQwMCIgaGV
pZ2h0PSI1MCIgcng9IiIgbWFya2VyLWVuZD0idXJsKCNkdEFycm93KSIvPgoKICA8IS0tIFN0ZXAg
b3cpIi8+CiAgPGNpcmNsZSBjeD0iMjMwIiBjeT0iMTU1IiByPSIxNSIgZmlsbD0icmdiYSgyNTUsM
jU1LDI1NSwwLjIpIi8+CiAgPHRleHQgeD0iMjMwIiB5PSIxNjEiIGZvbnQtZmFtaWx5PSJzeXN0ZW
0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTQiIG
ZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj45PC90ZXh0PgogIDx0ZXh0IHg9IjMwIiB5PSIyMDAiIHdpZHRoPSIzMDAiIGhlaWdodD0iNDUiIHJ4PSI4IiBmaWxsPSJ1cmwoI3Byb2JsZW1HcmFkKSIgZmlsdGVyPSJ1cmwoI2R0U2hhZG93KSIvPgoKICA8IS0tIFByb2JsZW0gQm94IC0tPgogIDxyZWN0IHg9IjIxZjVmOSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+RGVjaXNpb24gVHJlZS
AyOiBQYXJzaW5nIEZhaWx1cmVzPC90ZXh0PgoKICA8IS0tIFByb2JsZW0gQm94IC0tPgogIDxyZWN
0IHg9IjQwIiB5PSIzMDAiIHdpZHRoPSIzMDAiIGhlaWdodD0iNDUiIHJ4PSI4IiBmaWxsPSJ1cmwo
I3Byb2JsZW1HcmFkKSIgZmlsdGVyPSJ1cmwoI2R0U2hhZG93KSIvPgogIDx0ZXh0IHg9IjQwMCIge
T0iODMiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTQiIG
ZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj4xMDwvdGV4dD4KCiAgPCEtLSBQcm
```

AiIGhlaWdodD0iMzUiIHJ4PSI2IiBmaWxsPSJ1cmwoI3llc0dyYWQpIi8+CiAgPHRleHQgeD0iMTE
1IiB5PSIyNDIiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0i
MTEiIGZpbGw9IiNmZmYiIHRleHQtYW5jaG9yPSJtaWRkbGUiPkdvIHRvIFN0ZXAgMiDihpM8L3Rle
HQ+CgogIDwhLS0gTm8gUGF0aCAtIEZpeCBCb3ggLS0+CiAgPHJlY3QgeD0iNjIwIiB5PSIxOTUiIH
dpZHRoPSIxNjAiIGhlaWdodD0iOTAiIHJ4PSI2IiBmaWxsPSJ1cmwoI2ZpeEdyYWQpIiBmaWx0ZXI
9InVybCgjZHRTaGFkb3cpIi8+CiAgPHRleHQgeD0iNzAwIiB5PSIyMTUiIGZvbnQtZmFtaWx5PSJz
eXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZvbnQtd2VpZ2h0PSJib2xkIiBma
WxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5QYXJzZXIgTm90IFJ1bm5pbmc8L3RleHQ+Ci
AgPHRleHQgeD0iNjMwIiB5PSIyMzUiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWY
iIGZvbnQtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSI+4oCiIFBpcGVsaW5l
IG9yZGVyIGNvcnJlY3Q/PC90ZXh0PgogIDx0ZXh0IHg9IjYzMCIgeT0iMjUwIiBmb250LWZhbWlse
T0ic3lzdGVtLXVpLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNT
UsMjU1LDAuOSkiPuKAoiBQcm9jZXNzb3IgZW5hYmxlZD88L3RleHQ+CiAgPHRleHQgeD0iNjMwIiB
5PSIyNjUiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAi
IGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSI+4oCiIENvbmRpdGlvbiBtYXRjaGluZz88L3Rle
HQ+CiAgPHRleHQgeD0iNjMwIiB5PSIyODAiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2
VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiNmY2QzNGQiPuKGkiBFbmFibGUgJmFtcDsgcmVvcmR
lcjwvdGV4dD4KCiAgPCEtLSBTdGVwIDIgLS0+CiAgPGxpbmUgeDE9IjExNSIgeTE9IjI1NSIgeDI9
IjExNSIgeTI9IjI5NSIgc3Ryb2tlPSIjNjQ3NDhiIiBzdHJva2Utd2lkdGg9IjIiIG1hcmtlci1lb
mQ9InVybCgjZHRBcnJvdykiLz4KICA8cmVjdCB4PSIzMCIgeT0iMzAwIiB3aWR0aD0iMzgwIiBoZW
lnaHQ9IjUwIiByeD0iOCIgZmlsbD0idXJsKCNzdGVwR3JhZCkiIGZpbHRlcj0idXJsKCNkdFNoYWR
vdykiLz4KICA8Y2lyY2xlIGN4PSI2MCIgY3k9IjMyNSIgcj0iMTUiIGZpbGw9InJnYmEoMjU1LDI1
NSwyNTUsMC4yKSIvPgogIDx0ZXh0IHg9IjYwIiB5PSIzMzEiIGZvbnQtZmFtaWx5PSJzeXN0ZW0td
WksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTQiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjZm
ZmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj4yPC90ZXh0PgogIDx0ZXh0IHg9IjIzMCIgeT0iMzIwIiB
mb250LWZhbWlseT0ic3lzdGVtLXVpLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEzIiBmb250LXdl
aWdodD0iYm9sZCIgZmlsbD0iI2ZmZiIgdGV4dC1hbmNob3I9Im1pZGRsZSI+RG9lcyBEUEEgcGF0d
GVybiBtYXRjaCBsb2cgZm9ybWF0PzwvdGV4dD4KICA8dGV4dCB4PSIyMzAiIHk9IjMzOCIgZm9udC
1mYW1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSw
wLjgpIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5UZXN0IHBhdHRlcm4gd2l0aCBzYW1wbGUgbG9nIGlu
IERRTDwvdGV4dD4KCiAgPCEtLSBTdGVwIDIgQnJhbmNoZXMgLS0+CiAgPGxpbmUgeDE9IjEzMCIge
TE9IjM1MCIgeDI9IjEzMCIgeTI9IjM4MCIgc3Ryb2tlPSIjMTBiOTgxIiBzdHJva2Utd2lkdGg9Ij
IiLz4KICA8bGluZSB4MT0iMTMwIiB5MT0iMzgwIiB4Mj0iNjAiIHkyPSIzODAiIHN0cm9rZT0iIzE
wYjk4MSIgc3Ryb2tlLXdpZHRoPSIyIiBtYXJrZXItZW5kPSJ1cmwoI3llc0Fycm93KSIvPgogIDx0
ZXh0IHg9Ijk1IiB5PSIzNzUiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvb
nQtc2l6ZT0iMTEiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjMTBiOTgxIj7inJM8L3RleHQ+Cg
ogIDxsaW5lIHgxPSIyOTAiIHkxPSIzNTAiIHgyPSIyOTAiIHkyPSIzODAiIHN0cm9rZT0iI2Y1OWU
wYiIgc3Ryb2tlLXdpZHRoPSIyIi8+CiAgPGxpbmUgeDE9IjI5MCIgeTE9IjM4MCIgeDI9IjM2MCIg
eTI9IjM4MCIgc3Ryb2tlPSIjZjU5ZTBiIiBzdHJva2Utd2lkdGg9IjIiIG1hcmtlci1lbmQ9InVyb
Cgjbm9BcnJvdykiLz4KICA8dGV4dCB4PSIzMjUiIHk9IjM3NSIgZm9udC1mYW1pbHk9InN5c3RlbS
11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IiN
mNTllMGIiPuKclzwvdGV4dD4KCiAgPCEtLSBZZXMgLSBHbyB0byBTdGVwIDMgLS0+CiAgPHJlY3Qg
eD0iMjAiIHk9IjM5MCIgd2lkdGg9IjgwIiBoZWlnaHQ9IjMwIiByeD0iNiIgZmlsbD0idXJsKCN5Z
XNHcmFkKSIvPgogIDx0ZXh0IHg9IjYwIiB5PSI0MTAiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIH
NhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiNmZmYiIHRleHQtYW5jaG9yPSJtaWRkbGU
iPlN0ZXAgMyDihpM8L3RleHQ+CgogIDwhLS0gTm8gLSBQYXR0ZXJuIElzc3VlIC0tPgogIDxyZWN0
IHg9IjM2MCIgeT0iMzY1IiB3aWR0aD0iMTYwIiBoZWlnaHQ9Ijg1IiByeD0iNiIgZmlsbD0idXJsK
CNmaXhHcmFkKSIgZmlsdGVyPSJ1cmwoI2R0U2hhZG93KSIvPgogIDx0ZXh0IHg9IjQ0MCIgeT0iMz

```
g1IiBmb250LWZhbWlseT0ic3lzdGVtLXVpLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjExIiBmb25
0LXdlaWdodD0iYm9sZCIgZmlsbD0iI2ZmZiIgdGV4dC1hbmNob3I9Im1pZGRsZSI+UGF0dGVybiBJ
c3N1ZTwvdGV4dD4KICA8dGV4dCB4PSIzNzAiIHk9IjQwMiIgZm9udC1mYW1pbHk9InN5c3RlbS11a
Swgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSwwLjkpIj
7igKIgTG9nIGZvcm1hdCBjaGFuZ2VkPzwvdGV4dD4KICA8dGV4dCB4PSIzNzAiIHk9IjQxNyIgZm9
udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdi
YSgyNTUsMjU1LDI1NSwwLjkpIj7igKIgRXh0cmEgd2hpdGVzcGFjZT88L3RleHQ+CiAgPHRleHQge
D0iMzcwIiB5PSI0MzIiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2
l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSI+4oCiIE9wdGlvbmFsIGZpZWxkcz8
8L3RleHQ+CiAgPHRleHQgeD0iMzcwIiB5PSI0NDciIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNh
bnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiNmY2QzNGQiPuKGkiBVcGRhdGUgRFBMIHBhd
HRlcm48L3RleHQ+CgogIDwhLS0gU3RlcCAzIC0tPgogIDxsaW5lIHgxPSI2MCIgeTE9IjQyMCIgeD
I9IjYwIiB5Mj0iNDYwIiBzdHJva2U9IiM2NDc0OGIiIHN0cm9rZS13aWR0aD0iMiIgbWFya2VyLWV
uZD0idXJsKCNkdEFycm93KSIvPgogIDxyZWN0IHg9IjMwIiB5PSI0NjUiIHdpZHRoPSIzNDAiIGhl
aWdodD0iNTAiIHJ4PSI4IiBmaWxsPSJ1cmwoI3N0ZXBHcmFkKSIgZmlsdGVyPSJ1cmwoI2R0U2hhZ
G93KSIvPgogIDxjaXJjbGUgY3g9IjYwIiBjeT0iNDkwIiByPSIxNSIgZmlsbD0icmdiYSgyNTUsMj
U1LDI1NSwwLjIpIi8+CiAgPHRleHQgeD0iNjAiIHk9IjQ5NiIgZm9udC1mYW1pbHk9InN5c3RlbS1
1aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxNCIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IiNm
ZmYiIHRleHQtYW5jaG9yPSJtaWRkbGUiPjM8L3RleHQ+CiAgPHRleHQgeD0iMjEwIiB5PSI0ODUiI
GZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTMiIGZvbnQtd2
VpZ2h0PSJib2xkIiBmaWxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5BcmUgZmllbGQgbmF
tZXMgdmFsaWQ/PC90ZXh0PgogIDx0ZXh0IHg9IjIxMCIgeT0iNTAzIiBmb250LWZhbWlseT0ibW9u
b3NwYWNlIiBmb250LXNpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOCkiIHRleHQtY
W5jaG9yPSJtaWRkbGUiPkNoZWNrIHJlc2VydmVkIHdvcmRzLCBzcGVjaWFsIGNoYXJzPC90ZXh0Pg
oKICA8IS0tIFN0ZXAgMyBCcmFuY2hlcyAtLT4KICA8bGluZSB4MT0iMTMwIiB5MT0iNTE1IiB4Mj0
iMTMwIiB5Mj0iNTQ1IiBzdHJva2U9IiMxMGI5ODEiIHN0cm9rZS13aWR0aD0iMiIvPgogIDxsaW5l
IHgxPSIxMzAiIHkxPSI1NDUiIHgyPSI2MCIgeTI9IjU0NSIgc3Ryb2tlPSIjMTBiOTgxIiBzdHJva
2Utd2lkdGg9IjIiIG1hcmtlci1lbmQ9InVybCgjeWVzQXJyb3cpIi8+CiAgPHRleHQgeD0iOTUiIH
k9IjU0MCIgZm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSI
gZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IiMxMGI5ODEiPuKckzwvdGV4dD4KCiAgPGxpbmUgeDE9
IjI1MCIgeTE9IjUxNSIgeDI9IjI1MCIgeTI9IjU0NSIgc3Ryb2tlPSIjZjU5ZTBiIiBzdHJva2Utd
2lkdGg9IjIiLz4KICA8bGluZSB4MT0iMjUwIiB5MT0iNTQ1IiB4Mj0iMzIwIiB5Mj0iNTQ1IiBzdH
Jva2U9IiNmNTllMGIiIHN0cm9rZS13aWR0aD0iMiIgbWFya2VyLWVuZD0idXJsKCNub0Fycm93KSI
vPgogIDx0ZXh0IHg9IjI4NSIgeT0iNTQwIiBmb250LWZhbWlseT0ic3lzdGVtLXVpLCBzYW5zLXNl
cmlmIiBmb250LXNpemU9IjExIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0iI2Y1OWUwYiI+4pyXP
C90ZXh0PgoKICA8IS0tIFllcyAtIEdvIHRvIFN0ZXAgNCAtLT4KICA8cmVjdCB4PSIyMCIgeT0iNT
U1IiB3aWR0aD0iODAiIGhlaWdodD0iMzAiIHJ4PSI2IiBmaWxsPSJ1cmwoI3llc0dyYWQpIi8+CiA
gPHRleHQgeD0iNjAiIHk9IjU3NSIgZm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIg
Zm9udC1zaXplPSIxMCIgZmlsbD0iI2ZmZiIgdGV4dC1hbmNob3I9Im1pZGRsZSI+U3RlcCA0IOKGk
zwvdGV4dD4KCiAgPCEtLSBObyAtIE5hbWluZyBJc3N1ZSAtLT4KICA8cmVjdCB4PSIzMjAiIHk9Ij
UzMCIgd2lkdGg9IjE1NSIgaGVpZ2h0PSI3MCIgcng9IjYiIGZpbGw9InVybCgjZml4R3JhZCkiIGZ
pbHRlcj0idXJsKCNkdFNoYWRvdykiLz4KICA8dGV4dCB4PSIzOTciIHk9IjU1MCIgZm9udC1mYW1p
bHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZm9udC13ZWlnaHQ9ImJvb
GQiIGZpbGw9IiNmZmYiIHRleHQtYW5jaG9yPSJtaWRkbGUiPk5hbWluZyBJc3N1ZTwvdGV4dD4KIC
A8dGV4dCB4PSIzMzAiIHk9IjU2NyIgZm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiI
gZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSwwLjkpIj7igKIgVXNlIHVuZGVy
c2NvcmVzPC90ZXh0PgogIDx0ZXh0IHg9IjMzMCIgeT0iNTgyIiBmb250LWZhbWlseT0ic3lzdGVtL
XVpLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOS
```

kiPuKAoiBBdm9pZCByZXNlcnZlZCB3b3JkczwvdGV4dD4KICA8dGV4dCB4PSIzMzAiIHk9IjU5NyI
gZm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0i
I2ZjZDM0ZCI+4oaSIFJlbmFtZSBmaWVsZHM8L3RleHQ+CgogIDwhLS0gU3RlcCA0IC0tPgogIDxsa
W5lIHgxPSI2MCIgeTE9IjU4NSIgeDI9IjYwIiB5Mj0iNjA1IiBzdHJva2U9IiM2NDc0OGIiHN0cm
9rZS13aWR0aD0iMiIgbWFya2VyLWVuZD0idXJsKCNkdEFycm93KSIvPgogIDxyZWN0IHg9IjMwIiB
5PSI2MTAiIHdpZHRoPSIzNDAiIGhlaWdodD0iMzAiIHJ4PSI4IiBmaWxsPSJ1cmwoI3N0ZXBHcmFk
KSIgZmlsdGVyPSJ1cmwoI2R0U2hhZG93KSIvPgogIDxjaXJjbGUgY3g9IiIxMiIgY3k9IjYwIiBjeT0iMjA
SIxMiIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSwwLjIpIi8+CiAgPHRleHQgeD0iNjAiIHk9IjYzMC
IgZm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMiIgZm9udC1
3ZWlnaHQ9ImJvbGQiIGZpbGw9IiNmZmYiIHRleHQtYW5jaG9yPSJtaWRkbGUiPjQ8L3RleHQ+CiAg
PHRleHQgeD0iMjEwIiB5PSI2MzAiIHRleHQtYW5jaG9yPSJtaWRkbGUiPjU1MCIgeT0iNjMwIiB
3ZWlnaHQ9IjQwMCIgZmlsbD0iIzk0YTNiOCIgZm9udC1zaXplPSIxMiIgZm9udC1mYW1pbHk9InN
5c3RlbS11aSwgc2Fucy1zZXJpZiI+NTwvdGV4dD4KICA8dGV4dCB4PSIzNTAiIHk9IjYzMCIgZ
m9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iI2
ZmZiIgdGV4dC1hbmNob3I9Im1pZGRsZSI+4pyIEFkZCB0eXBlIGNvbnZlcnNpb248L3RleHQ+Cgo
gIDwhLS0gQXJyb3dzIHRvIGZpbmFsIGJveGVzIC0tPgogIDxsaW5lIHgxPSIzNzAiIHkxPSI2MjUi
IHgyPSIzOTUiIHk1PSI2MjUiIHN0cm9rZT0iIzEwYjk4MSIgc3Ryb2tlLXdpZHRoPSIyIiBtYXJrZ
XItZW5kPSJ1cmwoI3llc0Fycm93KSIvPgogIDxsaW5lIHgxPSI1MDAiIHkxPSI2MjUiIHgyPSI1MT
UiIHkyPSI2MjUiIHN0cm9rZT0iI2Y1OWUwYiIgc3Ryb2tlLXdpZHRoPSIyIiBtYXJrZXItZW5kPSJ
1cmwoI25vQXJyb3cpIi8+CjwvZnVnPg=)

Decision Tree 3: Masking Not Working

![Masking Decision Tree]
(
ciIHZpZXdCb3g9IjAgMCA4MDAgNjIwIj4KICA8ZGVmcz4KICAgIDxsaW5lYXJHcmFkaWVudCBpZD0
ibVByb2JsZW1HcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAlIj4KICAgICAg
PHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6I2VmNDQ0NDtzdG9wLW9wYWNpdHk6M
SIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojZGMyNjI2O3
N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5lYXJHcmFkaWV
udCBpZD0ibVN0ZXBHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAlIj4KICAg
ICAgPHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6IzNiODJmNjtzdG9wLW9wYWNpd
Hk6MSIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojMjU2M2
ViO3N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5lYXJHcmF
kaWVudCBpZD0ibVllc0dyYWQiIHgxPSIwJSIgeTE9IjAlIiB4Mj0iMTAwJSIgeTI9IjEwMCUiPgog
ICAgICA8c3RvcCBvZmZzZXQ9IjAlIiBzdHlsZT0ic3RvcC1jb2xvcjojMTBiOTgxO3N0b3Atb3BhY
2l0eToxIiAvPgogICAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiMwNT
k2Njk7c3RvcC1vcGFjaXR5OjEiIC8+CiAgICA8L2xpbmVhckdyYWRpZW50PgogICAgPGxpbmVhckdyYWRpZW50PgogICAgPG

```
yYWRpZW50IGlkPSJtTm9HcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAlIj4K
ICAgICAgPHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6I2Y1OWUwYjtzdG9wLW9wY
WNpdHk6MSIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojZD
k3NzA2O3N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5lYXJ
HcmFkaWVudCBpZD0ibUZpeEdyYWQiIHgxPSIwJSIgeTE9IjAlIiB4Mj0iMTAwJSIgeTI9IjEwMCUi
PgogICAgICA8c3RvcCBvZmZzZXQ9IjAlIiBzdHlsZT0ic3RvcC1jb2xvcjojOGI1Y2Y2O3N0b3Atb
3BhY2l0eToxIiAvPgogICAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eWxlPSJzdG9wLWNvbG9yOi
M3YzNhZWQ7c3RvcC1vcGFjaXR5OjEiIC8+CiAgICA8L2xpbmVhckdyYWRpZW50PgogICAgPGxpbmV
hckdyYWRpZW50IGlkPSJjcml0aWNhbEdyYWQiIHgxPSIwJSIgeTE9IjAlIiB4Mj0iMTAwJSIgeTI9
IjEwMCUiPgogICAgICA8c3RvcCBvZmZzZXQ9IjAlIiBzdHlsZT0ic3RvcC1jb2xvcjojZGMyNjI2O
3N0b3Atb3BhY2l0eToxIiAvPgogICAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eWxlPSJzdG9wLW
NvbG9yOiM5OTFiMWI7c3RvcC1vcGFjaXR5OjEiIC8+CiAgICA8L2xpbmVhckdyYWRpZW50PgogICA
gPGZpbHRlciBpZD0ibVNoYWRvdyI+CiAgICAgIDxmZURyb3BTaGFkb3cgZHg9IjIgIGR5PSIyIiBz
dGREZXZpYXRpb249IjMiIGZsb29kLW9wYWNpdHk9IjAuMTUiLz4KICAgIDwvZmlsdGVyPgogICAgP
G1hcmtlciBpZD0ibUFycm93IiBtYXJrZXJXaWR0aD0iMTAiIG1hcmtlckhlaWdodD0iNyIgcmVmWD
0iOSIgcmVmWT0iMy41IiBvcmllbnQ9ImF1dG8iPgogICAgICA8cG9seWdvbiBwb2ludHM9IjAgMCw
gMTAgMy41LCAwIDciIGZpbGw9IiM2NDc0OGIiLz4KICAgIDwvbWFya2VyPgogICAgPG1hcmtlciBp
ZD0ibVllc0Fycm93IiBtYXJrZXJXaWR0aD0iNyIgbWFya2VyVmlld0JveD0iMCAwIDEwIDciIGVsc
mVmWT0iMy41IiBvcmllbnQ9ImF1dG8iPgogICAgICA8cG9seWdvbiBwb2ludHM9IjAgMCwgMTAgMy
41LCAwIDciIGZpbGw9IiMxMGI5ODEiLz4KICAgIDwvbWFya2VyPgogICAgPG1hcmtlciBpZD0ibU5
vQXJyb3ciIG1hcmtlcldpZHRoPSIxMCIgbWFya2VySGVpZ2h0PSI3IiByZWZYPSI5IiByZWZZPSIz
LjUiIG9yaWVudD0iYXV0byI+CiAgICAgIDxwb2x5Z29uIHBvaW50cz0iMCAwLCAxMCAzLjUsIDAgN
yIgZmlsbD0iI2Y1OWUwYiIvPgogICAgPC9tYXJrZXI+CiAgPC9kZWZzPgoKICA8S0tIEJhY2tncm
91bmQgLS0+CiAgPHJlY3Qgd2lkdGg9IjgwMCIgaGVpZ2h0PSI2MjAiIGZpbGw9IiMwZjE3MmEiIHJ
4PSIxMCIvPgoKICA8S0tIFRpdGxlIC0tPgogIDx0ZXh0IHg9IjMzUiIGZvbnQtZmFt
aWx5PSJzeXN0ZW0tdWksIC1hcHBsZS1zeXN0ZW0sIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMjAiI
GZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjZjFmNWY5IiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5EZW
Npc2lvbiBUcmVlIDM6IE1hc2tpbmcgTm90IFdvcmtpbmc8L3RleHQ+CgogIDwhLS0gUHJvYmxlbSB
Cb3ggLSBDcml0aWNhbCAtLT4KICA8cmVjdCB4PSIyMDAiIHk9IjU1IiB3aWR0aD0iNDAwIiBoZWln
aHQ9IjUwIiByeD0iOCIgZmlsbD0idXJsKCNjcml0aWNhbEdyYWQpIiBmaWx0ZXI9InVybCgjbVNoY
WRvdykiLz4KICA8dGV4dCB4PSIyMjAiIHk9Ijc1IiBmb250LWZhbWlseT0ic3lzdGVtLXVpLCBzYW
5zLXNlcmlmIiBmb250LXNpemU9IjEyIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0iI2ZlY2FjYSI
+4pqg77iPIENSSVRJQ0FMPC90ZXh0PgogIDx0ZXh0IHg9IjQwMCIgeT0iOTMiIGZvbnQtZmFtaWx5
PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTMiIGZvbnQtd2VpZ2h0PSJib2xkI
iBmaWxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5TZW5zaXRpdmUgZGF0YSAoUElJL1BISS
9DSEQpIHZpc2libGUgaW4gc3RvcmVkIGxvZ3M8L3RleHQ+CgogIDxLLS0gQXJyb3cgdG8gU3RlcCA
xIC0tPgogIDxsaW5lIHgxPSI0MDAiIHkxPSIxMDUiIHgyPSI0MDAiIHkyPSIxMzAiIHN0cm9rZT0i
IzY0NzQ4YiIgc3Ryb2tlLXdpZHRoPSIyIiBtYXJrZXItZW5kPSJ1cmwoI21BcnJvdykiLz4KICAgP
CEtLSBTdGVwIDEgLS0+CiAgPHJlY3QgeD0iMTgwIiB5PSIxMzUiIHdpZHRoPSI0NDAiIGhlaWdodD
0iNTUiIHJ4PSI4IiBmaWxsPSJ1cmwoI21TdGVwR3JhZCkiIGZpbHRlcj0idXJsKCNtU2hhZG93KSI
vPgogIDxjaXJjbGUgY3g9IjIxMCIgY3k9IjE2MiIgcj0iMTUiIGZpbGw9InJnYmEoMjU1LDI1NSwy
NTUsMC4yKSIvPgogIDx0ZXh0IHg9IjIxMCIgeT0iMTY4IiBmb250LWZhbWlseT0ic3lzdGVtLXVpL
CBzYW5zLXNlcmlmIiBmb250LXNpemU9IjE0IiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0iI2ZmZi
IgdGV4dC1hbmNob3I9Im1pZGRsZSI+MTwvdGV4dD4KICA8dGV4dCB4PSIyNDAiIHk9IjE1NSIgZm9
udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIyMyIgZm9udC13ZWln
aHQ9ImJvbGQiIGZpbGw9IiNmZmYiIHRleHQtYW5jaG9yPSJtaWRkbGUiPklzIHRoZSBtYXNraW5nI
HByb2Nlc3NvciBydW5uaW5nPzwvdGV4dD4KICA8dGV4dCB4PSIyNDAiIHk9IjE3NSIgZm9udC1mYW
1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSwwLjg
```

```
pIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5RdWVyeTogZmV0Y2ggbG9ncyB8IGZpbHRlciBjb250YWlu
cyhjb250ZW50LCAiNDExMSIpIHwgbGltaXQgMTwvdGV4dD4KCiAgPCEtLSBTdGVwIDEgQnJhbmNoZ
XMgLS0+CiAgPGxpbmUgeDE9IjI4MCIgeTE9IjE5MCIgeDI9IjI4MCIgeTI9IjIyMCIgc3Ryb2tlPS
IjZjU5ZTBiIiBzdHJva2Utd2lkdGg9IjIiLz4KICA8bGluZSB4MT0iMjgwIiB5MT0iMjIwIiB4Mj0
iMTUwIiB5Mj0iMjIwIiBzdHJva2U9IiNmNTllMGIiIHN0cm9rZS13aWR0aD0iMiIgbWFya2VyLWVu
ZD0idXJsKCNtTm9BcnJvdykiLz4KICA8dGV4dCB4PSIyMTUiIHk9IjIxNSIgZm9udC1mYW1pbHk9I
nN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZm9udC13ZWlnaHQ9ImJvbGQiIG
ZpbGw9IiNmNTllMGIiPkZPVU5EIOKclzwvdGV4dD4KICAgPGxpbmUgeDE9IjUyMCIgeTE9IjE5MCI
geDI9IjUyMCIgeTI9IjIyMCIgc3Ryb2tlPSIjMTBiOTgxIiBzdHJva2Utd2lkdGg9IjIiLz4KICA8
bGluZSB4MT0iNTIwIiB5MT0iMjIwIiB4Mj0iNjUwIiB5Mj0iMjIwIiBzdHJva2U9IiMxMGI5ODEiI
HN0cm9rZS13aWR0aD0iMiIgbWFya2VyLWVuZD0idXJsKCNtWWVzQXJyb3cpIi8+CiAgPHRleHQgeD
0iNTg1IiB5PSIyMTUiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l
6ZT0iMTEiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjMTBiOTgxIj5OT1QgRk9VTkQg4pyTPC90
ZXh0PgoKICA8IS0tIEZvdW5kID0gTWFza2luZyBOT1Qgd29ya2luZywgZ28gdG8gU3RlcCAyIC0tP
gogIDxyZWN0IHg9IjQwIiB5PSIyMzAiIHdpZHRoPSIxNTAiIGhlaWdodD0iMzUiIHJ4PSI2IiBmaW
xsPSJ1cmwoI21Ob0dyYWQpIi8+CiAgPHRleHQgeD0iMTE1IiB5PSIyNTIiIGZvbnQtZmFtaWx5PSJ
zeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZpbGw9IiNmZmYiIHRleHQtYW5j
aG9yPSJtaWRkbGUiPk1hc2tpbmcgYnJva2VuIOKGkiBTdGVwIDI8L3RleHQ+CgogIDwhLS0gTm90I
EZvdW5kID0gTWF5IGJlIHdvcmtpbmcgLS0+CiAgPHJlY3QgeD0iNjIwIiB5PSIyMzAiIHdpZHRoPS
IxNjAiIGhlaWdodD0iNTAiIHJ4PSI2IiBmaWxsPSJ1cmwoI21ZZXNHcmFkKSIvPgogIDx0ZXh0IHg
9IjcwMCIgeT0iMjUwIiBmb250LWZhbWlseT0ic3lzdGVtLXVpLCBzYW5zLXNlcmlmIiBmb250LXNp
emU9IjExIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0iI2ZmZiIgdGV4dC1hbmNob3I9Im1pZGRsZ
SI+TWF5IGJlIHdvcmtpbmc8L3RleHQ+CiAgPHRleHQgeD0iNzAwIiB5PSIyNjgiIGZvbnQtZmFtaW
x5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI
1NSwyNTUsMC45KSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+VmVyaWZ5IHdpdGgga25vd24gdGVzdCBk
YXRhPC90ZXh0PgoKICA8IS0tIFN0ZXAgMiAtLT4KICA8bGluZSB4MT0iMTE1IiB5MT0iMjY1IiB4M
j0iMTE1IiB5Mj0iMzAwIiBzdHJva2U9IiM2NDc0OGIiIHN0cm9rZS13aWR0aD0iMiIgbWFya2VyLW
VuZD0idXJsKCNtQXJyb3cpIi8+CiAgPHJlY3QgeD0iNDAiIHk9IjMwNSIgd2lkdGg9IjIyQwMCIgaGV
pZ2h0PSI1NSIgcng9IjgiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC4wNSkiIHN0cm9rZT0icmdi
KDI1NSwyNTUsMjU1LDAuMikiLz4KICA8dGV4dCB4PSIxMTUiIHk9IjMyNSIgZm9udC1mYW1pbHk9I
nN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMyIgZm9udC13ZWlnaHQ9IjYwMCIgZm
lsbD0icmdiKDI1NSwyNTUsMjU1KSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+MTUsMjU1LDAuMikiLz4
KICA8dGV4dCB4PSIxMTUiIHk9IjM0MCIgZm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJp
ZiIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiKDE1MywyMjIsMjU1KSIgdGV4dC1hbmNob3I9Im1pZ
GRsZSI+MTUsMjU1LDAuMikiLz4KICA8dGV4dCB4PSIxMTUiIHk9IjM1NSIgZm9udC1mYW1pbHk9Ik
NvdXJpZXIsIG1vbm9zcGFjZSIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiKDE1MywyMjIsMjU1KSI
gdGV4dC1hbmNob3I9Im1pZGRsZSI+MjU1LDAuMikiLz4KICA8dGV4dCB4PSIxMTUiIHk9IjM3MCIg
Zm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMyIgZm9udC13Z
WlnaHQ9IjYwMCIgZmlsbD0icmdiKDI1NSwyNTUsMjU1KSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+MT
UsMjU1LDAuMikiLz4KICA8dGV4dCB4PSIxMTUiIHk9IjM4NSIgZm9udC1mYW1pbHk9InN5c3RlbS1
pLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjExIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0iI2ZmZ
mZiIgdGV4dC1hbmNob3I9Im1pZGRsZSI+MjU1LDAuMikiLz4KICA8dGV4dCB4PSIxMTUiIHk9IjQw
MCIgZm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsb
GQiPk1hc2tpbmcgYnJva2VuIOKGkiBTdGVwIDI8L3RleHQ+CgogIDwhLS0gTm90IEZvdW5kID0gTW
F5IGJlIHdvcmtpbmcgLS0+CiAgPHJlY3QgeD0iNjIwIiB5PSIzMDUiIHdpZHRoPSIxNjAiIGhlaWd
odD0iNTAiIHJ4PSI2IiBmaWxsPSJ1cmwoI21ZZXNHcmFkKSIvPgogIDx0ZXh0IHg9IjcwMCIgeT
0iMzI1IiBmb250LWZhbWlseT0ic3lzdGVtLXVpLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjExIiB
maWxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5NYXkgYmUgYmUgbm90IGZvdW5kIOKGkiBT
dGVwIDI8L3RleHQ+CiAgPHRleHQgeD0iNzAwIiB5PSIzMjUiIGZvbnQtZmFtaWx5PSJzeXN0ZW0td
WksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KS
IgdGV4dC1hbmNob3I9Im1pZGRsZSI+VmVyaWZ5IHdpdGgga25vd24gdGVzdCBkYXRhPC90ZXh0Pgo
KICA8IS0tIFN0ZXAgMiAtLT4KICA8bGluZSB4MT0iMTE1IiB5MT0iMjY1IiB4Mj0iMTE1IiB5Mj0i
MzAwIiBzdHJva2U9IiM2NDc0OGIiIHN0cm9rZS13aWR0aD0iMiIgbWFya2VyLWVuZD0idXJsKCNtQ
XJyb3cpIi8+CiAgPHJlY3QgeD0iNDAiIHk9IjMwNSIgd2lkdGg9IjIyMCIgaGVpZ2h0PSI1NSIgcn
g9IjgiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC4wNSkiIHN0cm9rZT0icmdiKDI1NSwyNTUsMjU
1LDAuMikiLz4KICA8dGV4dCB4PSIxMTUiIHk9IjMyNSIgZm9udC1mYW1pbHk9InN5c3RlbS11aSwg
c2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMyIgZm9udC13ZWlnaHQ9IjYwMCIgZmlsbD0icmdiKDI1N
SwyNTUsMjU1KSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+MTUsMjU1LDAuMikiLz4KICA8dGV4dCB4PS
IxMTUiIHk9IjM0MCIgZm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXp
lPSIxMCIgZmlsbD0icmdiKDE1MywyMjIsMjU1KSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+MjU1LDAu
MikiLz4KICA8dGV4dCB4PSIxMTUiIHk9IjM3MCIgZm9udC1mYW1pbHk9Ik5vdG8gU2FucyBTeW1
iLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEzIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0icmdi
KDI1NSwyNTUsMjU1KSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+MTUsMjU1LDAuMikiLz4KICA8dGV4d
CB4PSIxMTUiIHk9IjM4NSIgZm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC
1zaXplPSIxMSIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC4yKSI
gdGV4dC1hbmNob3I9Im1pZGRsZSI+MjU1LDAuMikiLz4KICA8dGV4dCB4PSIxMTUiIHk9IjQwMCIg
Zm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iI
HJnYmEoMjU1LDI1NSwyNTUsMC4yKSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+MjU1LDAuMikiLz4KPC
9zdmc+Cg==
```

iIHk9IjM4NSIgZm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIx
MSIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IiNmNTllMGIiPuKclzwvdGV4dD4KCiAgPCEtLSBZZ
XMgLSBHbyB0byBTdGVwIDMgLS0+CiAgPHJlY3QgeD0iMjUiIHk9IjQwMCIgd2lkdGg9IjgwIiBoZW
lnaHQ9IjMwIiByeD0iNiIgZmlsbD0idXJsKCNtWWVzR3JhZCkiLz4KICA8dGV4dCB4PSI2NSIgeT0
iNDIwIiBmb250LWZhbWlseT0ic3lzdGVtLXVpLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBm
aWxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5TdGVwIDMg4oaTPC90ZXh0PgoKICA8IS0tI
E5vIC0gT3JkZXIgSXNzdWUgLS0+CiAgPHJlY3QgeD0iNDAwIiB5PSIzNzUiIHdpZHRoPSIxODAiIG
hlaWdodD0iODUiIHJ4PSI2IiBmaWxsPSJ1cmwoI21GaXhHcmFkKSIgZmlsdGVyPSJ1cmwoI21TaGF
kb3cpIi8+CiAgPHRleHQgeD0iNDkwIiB5PSIzOTUiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNh
bnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjZmZmIiB0Z
Xh0LWFuY2hvcj0ibWlkZGxlIj5Qcm9jZXNzb3IgT3JkZXIgSXNzdWU8L3RleHQ+CiAgPHRleHQgeD
0iNDEwIiB5PSI0MTUiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l
6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSI+4oCiIE1vdmUgbWFza2luZyBlYXJs
aWVyPC90ZXh0PgogIDx0ZXh0IHg9IjQxMCIgeT0iNDMyIiBmb250LWZhbWlseT0ic3lzdGVtLXVpL
CBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiPu
KAoiBDb250ZW50IG1hc2tpbmcgRklSU1Q8L3RleHQ+CiAgPHRleHQgeD0iNDEwIiB5PSI0NDkiIGZ
vbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiNm
Y2QzNGQiPuKGkiBSZW9yZGVyICZhbXA7IHRlc3Q8L3RleHQ+CgogIDwhLS0gU3RlcCAzIC0tPgogI
DxsaW5lIHgxPSI2NSIgeTE9IjQzMCIgeDI9IjY1IiB5Mj0iNDY1IiBzdHJva2U9IiM2NDc0OGIiIH
N0cm9rZS13aWR0aD0iMiIgbWFya2VyLWVuZD0idXJsKCNtQXJyb3cpIi8+CiAgPHJlY3QgeD0iNDA
iIHk9IjQ3MCIgd2lkdGg9IjQwMCIgaGVpZ2h0PSI1NSIgcng9IjgiIGZpbGw9InVybCgjbVN0ZXBH
cmFkKSIgZmlsdGVyPSJ1cmwoI21TaGFkb3cpIi8+CiAgPGNpcmNsZSBjeD0iNzAiIGN5PSI0OTciI
HI9IjE1IiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuMikiLz4KICA8dGV4dCB4PSI3MCIgeT0iNT
AzIiBmb250LWZhbWlseT0ic3lzdGVtLXVpLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjE0IiBmb25
0LXdlaWdodD0iYm9sZCIgZmlsbD0iI2ZmZiIgdGV4dC1hbmNob3I9Im1pZGRsZSI+MzwvdGV4dD4K
ICA8dGV4dCB4PSIyNjAiIHk9IjQ5MCIgZm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZ
iIgZm9udC1zaXplPSIxMyIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IiNmZmYiIHRleHQtYW5jaG
9yPSJtaWRkbGUiPkRvZXMgdGhlIHJlZ2V4IHBhdHRlcm4gbWF0Y2g/PC90ZXh0PgogIDx0ZXh0IHg
9IjI2MCIgeT0iNTEwIiBmb250LWZhbWlseT0ibW9ub3NwYWNlIiBmb250LXNpemU9IjEwIiBmaWxs
PSJyZ2JhKDI1NSwyNTUsMjU1LDAuOCkiIHRleHQtYW5jaG9yPSJtaWRkbGUiPlRlc3Q6IHJlcGxhY
2VBbGwoY29udGVudCwgIlxcYlxcZHs0fS4uLlxcZHs0fVxcYiIsICJbUkVEQUNURURdIik8L3RleH
Q+CgogIDwhLS0gU3RlcCAzIEJyYW5jaGVzIC0tPgogIDxsaW5lIHgxPSIxNDAiIHkxPSI1MjUiIHg
yPSIxNDAiIHkyPSI1NTUiIHN0cm9rZT0iIzEwYjk4MSIgc3Ryb2tlLXdpZHRoPSIyIi8+CiAgPGxp
bmUgeDE9IjE0MCIgeTE9IjU1NSIgeDI9IjYwIiB5Mj0iNTU1IiBzdHJva2U9IiMxMGI5ODEiIHN0c
m9rZS13aWR0aD0iMiIgbWFya2VyLWVuZD0idXJsKCNtWWVzQXJyb3cpIi8+CiAgPHRleHQgeD0iMT
AwIiB5PSI1NTAiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0
iMTEiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjMTBiOTgxIj7inJM8L3RleHQ+CgogIDxsaW5l
IHgxPSIzMjAiIHkxPSI1MjUiIHgyPSIzMjAiIHkyPSI1NTUiIHN0cm9rZT0iI2Y1OWUwYiIgc3Ryb
2tlLXdpZHRoPSIyIi8+CiAgPGxpbmUgeDE9IjMyMCIgeTE9IjU1NSIgeDI9IjQwMCIgeTI9IjU1NS
Igc3Ryb2tlPSIjZjU5ZTBiIiBzdHJva2Utd2lkdGg9IjIiIG1hcmtlci1lbmQ9InVybCgjbU5vQXJ
yb3cpIi8+CiAgPHRleHQgeD0iMzYwIiB5PSI1NTAiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNh
bnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjZjU5ZTBiI
j7inJc8L3RleHQ+CgogIDwhLS0gWWVzIC0gR28gdG8gU3RlcCA0IC0tPgogIDxyZWN0IHg9IjI1Ii
B5PSI1NjUiIHdpZHRoPSI4MCIgaGVpZ2h0PSIzMCIgcng9IjYiIGZpbGw9InVybCgjbVllc0dyYWQ
pIi8+CiAgPHRleHQgeD0iNjUiIHk9IjU4NSIgZm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1z
ZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iI2ZmZiIgdGV4dC1hbmNob3I9Im1pZGRsZSI+U3Rlc
CA0IOKGkzwvdGV4dD4KCiAgPCEtLSBObyAtIFBhdHRlcm4gSXNzdWUgLS0+CiAgPHJlY3QgeD0iND
AwIiB5PSI1NDAiIHdpZHRoPSIxODAiIGhlaWdodD0iNzAiIHJ4PSI2IiBmaWxsPSJ1cmwoI21GaXh

```
HcmFkKSIgZmlsdGVyPSJ1cmwoI21TaGFkb3cpIi8+CiAgPHRleHQgeD0iNDkwIiB5PSI1NjAiIGZv
bnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZvbnQtd2VpZ
2h0PSJib2xkIiBmaWxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5QYXR0ZXJuIEIzc3VlPC
90ZXh0PgogIDx0ZXh0IHg9IjQxMCIgeT0iNTgwIiBmb250LWZhbWlseT0ic3lzdGVtLXVpLCBzYW5
zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiPuKAoiBF
c2NhcGluZyBjb3JyZWN0PyAoXFxkKTwvdGV4dD4KICA8dGV4dCB4PSI0MTAiIHk9IjU5NyIgZm9ud
C1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iI2ZjZD
M0ZCI+4oaSIFVwZGF0ZSByZWdleCBwYXR0ZXJuPC90ZXh0PgoKICA8IS0tIFN0ZXAgNCAtIEZpbmF
sIC0tPgogIDxyZWN0IHg9IjEyMCIgeT0iNTY1IiB3aWR0aD0iMjQwIiBoZWlnaHQ9IjQ1IiByeD0i
OCIgZmlsbD0idXJsKCNtU3RlcEdyYWQpIiBmaWx0ZXI9InVybCgjbVNoYWRvdykiLz4KICA8dGV4dC
2xlIGN4PSIxNDUiIGN5PSI1ODciIHI9IjEyIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuMikiLz
4KICA8dGV4dCB4PSIxNDUiIHk9IjU5MiIgZm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJ
pZiIgZm9udC1zaXplPSIxMiIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IiNmZmYiIHRleHQtYW5j
aG9yPSJtaWRkbGUiPjQ8L3RleHQ+CiAgPHRleHQgeD0iMjUwIiB5PSI1ODIiIGZvbnQtZmFtaWx5P
SJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTIiIGZvbnQtd2VpZ2h0PSJib2xkIi
BmaWxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5BbGwgZmllbGRzIG1hc2tlZD88L3RleHQ
+CiAgPHRleHQgeD0iMjUwIiB5PSI2MDAiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2Vy
aWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC44KSIgdGV4dC1hbmNob
3I9Im1pZGRsZSI+Q2hlY2sgQUxMIGZpZWxkcyBmb3IgbGVha3M8L3RleHQ+CgogIDwhLS0gRmluYW
wgb3V0Y29tZXMgLS0+CiAgPHJlY3QgeD0iNjAwIiB5PSI0NzAiIHdpZHRoPSIxNzAiIGhlaWdodD0
iNTUiIHJ4PSI2IiBmaWxsPSJ1cmwoI21ZZXNHcmFkKSIgZmlsdGVyPSJ1cmwoI21TaGFkb3cpIi8+
CiAgPHRleHQgeD0iNjg1IiB5PSI0OTMiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2Vya
WYiIGZvbnQtc2l6ZT0iMTIiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjZmZmIiB0ZXh0LWFuY2
hvcj0ibWlkZGxlIj7inJMgQWxsIE1hc2tlZDwvdGV4dD4KICA8dGV4dCB4PSI2ODUiIHk9IjUxMiI
gZm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0i
cmdiYSgyNTUsMjU1LDI1NSwwLjkpIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5Ob3cgYSBmYXNraW5nIGNvbXBsZX
RlITwvdGV4dD4KICAgPHRleHQgeD0iNjAwIiB5PSI1MzUiIGhlaWdodD0iNz
AiIHJ4PSI2IiBmaWxsPSJ1cmwoI2NyaXRpY2FsR3JhZCkiIGZpbHRlcj0idXJsKCNtU2hhZG93KSI
vPgogIDx0ZXh0IHg9IjY4NSIgeT0iNTU4IiBmb250LWZhbWlseT0ic3lzdGVtLXVpLCBzYW5zLXNl
cmlmIiBmb250LXNpemU9IjEyIiBmaWxsPSIjZmZmIiBmb250LXdlaWdodD0iYm9sZCIgdGV4dC1hb
mNob3I9Im1pZGRsZSI+4pyXIERhdGEgTGVha2VkPC90ZXh0PgogIDx0ZXh0IHg9IjY4NSIgeT0iNT
c4IiBmb250LWZhbWlseT0ic3lzdGVtLXVpLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWx
sPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiPuKAoiBBZGQgZmllbGQgbWFza2luZzwvdGV4dD4KICA8
dGV4dCB4PSI2ODUiIHk9IjU5NSIgZm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZ
m9udC1zaXplPSIxMCIgZmlsbD0iI2ZjZDM0ZCI+4oaSIFVzZSBmaWVsZHNSZW1vdmU8L3RleHQ+Cg
ogIDwhLS0gUGVyZm9ybWFuY2UgZyBsaW5lcyB0byBmaW5hbCBib3hlcyAtLT4KICA8bGluZSB4MT0iMzY
wIiB5MT0iNTg3IiB4Mj0iNTk1IiB5Mj0iNTAwIiBzdHJva2U9IiMxMGI5ODEiIHN0cm9rZS13aWR0
aD0iMiIgbWFya2VyLWVuZD0idXJsKCNtWWVzQXJyb3cpIi8+CiAgPGxpbmUgeDE9IjM2MCIgeTE9I
jU4NyIgeDI9IjU0NSIgeTI9IjU3MCIgc3Ryb2tlPSIjZjU5ZTBiIiBzdHJva2Utd2lkdGg9IjIiIG
1hcmtlci1lbmQ9InVybCgjbU5vQXJyb3cpIi8+CjwvZnVsbC1zdW1tYXJ5PgoKCgotLS0KCiMjIyBEZWNpc2lvbiBUcmVlIDQ6IFBlcmZvcm1hbmNlIFByb2JsZW1zCgohW1BlcmZvcm1hbmNlIERlY2lzaW9uIFRyZWVd
```

(
ciIHZpZXdCb3g9IjAgMCA4NTAgNzAwIj4KICA8ZGVmcz4KICAgIDxsaW5lYXJHcmFkaWVudCBpZD0
icFByb2JsZW1HcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAlIj4KICAgICAg
PHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6I2Y1OWUwYjtzdG9wLW9wYWNpdHk6M
SIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojZDk3NzA2O3
N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5lYXJHcmFkaWV
udCBpZD0icFN0ZXBHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAlIj4KICAg
ICAgPHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6IzNiODJmNjtzdG9wLW9wYWNpd
Hk6MSIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojMjU2M2
ViO3N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5lYXJHcmF
kaWVudCBpZD0icFNvbHV0aW9uR3JhZCIgeDE9IjAlIiB5MT0iMCUiIHgyPSIxMDAlIiB5Mj0iMTAw
JSI+CiAgICAgIDxzdG9wIG9mZnNldD0iMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiMxMGI5ODE7c3Rvc
C1vcGFjaXR5OjEiIC8+CiAgICAgIDxzdG9wIG9mZnNldD0iMTAwJSIgc3R5bGU9InN0b3AtY29sb3
I6IzA1OTY2OTtzdG9wLW9wYWNpdHk6MSIgLz4KICAgIDwvbGluZWFyR3JhZGllbnQ+CiAgICA8bGl
uZWFyR3JhZGllbnQgaWQ9InBDYXRlZ29yeUdyYWQiIHgxPSIwJSIgeTE9IjAlIiB4Mj0iMTAwJSIg
eTI9IjEwMCUiPgogICAgICA8c3RvcCBvZmZzZXQ9IjAlIiBzdHlsZT0ic3RvcC1jb2xvcjojOGI1Y
2Y2O3N0b3Atb3BhY2l0eToxIiAvPgogICAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eWxlPSJzdG
9wLWNvbG9yOiM3YzNhZWQ7c3RvcC1vcGFjaXR5OjEiIC8+CiAgICA8L2xpbmVhckdyYWRpZW50Pgo
gICAgPGxpbmVhckdyYWRpZW50IGlkPSJwV2FybmluZ0dyYWQiIHgxPSIwJSIgeTE9IjAlIiB4Mj0i
MTAwJSIgeTI9IjEwMCUiPgogICAgICA8c3RvcCBvZmZzZXQ9IjAlIiBzdHlsZT0ic3RvcC1jb2xvc
jojZWY0NDQ0O3N0b3Atb3BhY2l0eToxIiAvPgogICAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eW
xlPSJzdG9wLWNvbG9yOiNkYzI2MjY7c3RvcC1vcGFjaXR5OjEiIC8+CiAgICA8L2xpbmVhckdyYWR
pZW50PgogICAgPGZpbHRlciBpZD0icFNoYWRvdyI+CiAgICAgIDxmZURyb3BTaGFkb3cgZHg9IjIi
IGR5PSIyIiBzdGREZXZpYXRpb249IjMiIGZsb29kLW9wYWNpdHk9IjAuMTUiLz4KICAgIDwvZmlsd
GVyPgogICAgPG1hcmtlciBpZD0icEFycm93IiBtYXJrZXJXaWR0aD0iMTAiIG1hcmtlckhlaWdodD
0iNyIgcmVmWD0iOSIgcmVmWT0iMy41IiBvcmllbnQ9ImF1dG8iPgogICAgICA8cG9seWdvbiBwb2l
udHM9IjAgMCwgMTAgMy41LCAwIDciIGZpbGw9IiM2NDc0OGIiLz4KICAgIDwvbWFya2VyPgogIDwv
ZGVmcz4KCiAgPCEtLSBCYWNrZ3JvdW5kIC0tPgogIDxyZWN0IHdpZHRoPSI4NTAiIGhlaWdodD0iN
zAwIiBmaWxsPSIjMGYxNzJhIiByeD0iMTAiLz4KCiAgPCEtLSBUaXRsZSAtLT4KICA8dGV4dCB4PS
I0MjUiIHk9IjM1IiBmb250LWZhbWlseT0ic3lzdGVtLXVpLCAtYXBwbGUtc3lzdGVtLCBzYW5zLXN
lcmlmIiBmb250LXNpemU9IjIwIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0iI2YxZjVmOSIgdGV4
dC1hbmNob3I9Im1pZGRsZSI+RGVjaXNpb24gVHJlZSA0OiBQZXJmb3JtYW5jZSBQcm9ibGVtczwvd
GV4dD4KCiAgPCEtLSBQcm9ibGVtIEJveCAtLT4KICA8cmVjdCB4PSIyMjUiIHk9IjU1IiB3aWR0aD
0iNDAwIiBoZWlnaHQ9IjUwIiByeD0iOCIgZmlsbD0idXJsKCNwUHJvYmxlbUdyYWQpIiBmaWx0ZXI
9InVybCgjcFNoYWRvdykiLz4KICA8dGV4dCB4PSI0MjUiIHk9Ijc4IiBmb250LWZhbWlseT0ic3lz
dGVtLXVpLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjE0IiBmb250LXdlaWdodD0iYm9sZCIgZmlsb
D0iI2ZmZiIgdGV4dC1hbmNob3I9Im1pZGRsZSI+U2xvdyBwcm9jZXNzaW5nLCBkZWxheWVkIGxvZ3
MsIG9yIGhpZ2ggY29zdHM8L3RleHQ+CiAgPHRleHQgeD0iNDI1IiB5PSI5NSIgZm9udC1mYW1pbHk
9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZmlsbD0icmdiYSgyNTUsMjU1
LDI1NSwwLjgpIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5RdWVyeSBsYXRlbmN5ICZndDsgNSBtaW4g4
oCiIFBpcGVsaW5lIHRpbWUgJmd0OyAxMDBtcyDigKIgVGltZW91dHM8L3RleHQ+CgogIDwhLS0gQX
Jyb3cgdG8gU3RlcCAxIC0tPgogIDxsaW5lIHgxPSI0MjUiIHkxPSIxMDUiIHgyPSI0MjUiIHkyPSI
xMzAiIHN0cm9rZT0iIzY0NzQ4YiIgc3Ryb2tlLXdpZHRoPSIyIiBtYXJrZXItZW5kPSJ1cmwoI3BB
cnJvdykiLz4KCiAgPCEtLSBTdGVwIDE6IElkZW50aWZ5IEJvdHRsZW5lY2sgLS0+CiAgPHJlY3Qge
D0iMjI1IiB5PSIxMzUiIHdpZHRoPSI0MDAiIGhlaWdodD0iNTAiIHJ4PSI4IiBmaWxsPSJ1cmwoI3
BTdGVwR3JhZCkiIGZpbHRlcj0idXJsKCNwU2hhZG93KSIvPgogIDxjaXJjbGUgY3g9IjI1NSIgY3k
9IjE2MCIgcj0iMTUiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC4yKSIvPgogIDx0ZXh0IHg9IjI1
NSIgeT0iMTY2IiBmb250LWZhbWlseT0ic3lzdGVtLXVpLCBzYW5zLXNlcmlmIiBmb250LXNpemU9I

```
jE0IiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0iI2ZmZiIgdGV4dC1hbmNob3I9Im1pZGRsZSI+MT
wvdGV4dD4KICA8dGV4dCB4PSI0NTAiIHk9IjE1NSIgZm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2F
ucy1zZXJpZiIgZm9udC1zaXplPSIxNCIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IiNmZmYiIHRl
eHQtYW5jaG9yPSJtaWRkbGUiPklkZW50aWZ5IHRoZSBib3R0bGVuZWNrPC90ZXh0PgogIDx0ZXh0I
Hg9IjQ1MCIgeT0iMTczIiBmb250LWZhbWlseT0ic3lzdGVtLXVpLCBzYW5zLXNlcmlmIiBmb250LX
NpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOCkiIHRleHQtYW5jaG9yPSJtaWRkbGU
iPldoaWNoIHBpcGVsaW5lIGlzIHNsb3dlc3Q/IFdoYXQncyBjYXVzaW5nIGRlbGF5cz88L3RleHQ+
CgogIDwhLS0gV2FybmluZyBhYm91dCBOVUxMIGZpZWxkcyAtLT4KICA8cmVjdCB4PSI2NDAiIHk9I
jEzNSIgd2lkdGg9IjE5MCIgaGVpZ2h0PSI1MCIgcng9IjYiIGZpbGw9InVybCgjcFdhcm5pbmdHcm
FkKSIgb3BhY2l0eT0iMC45Ii8+CiAgPHRleHQgeD0iNzM1IiB5PSIxNTUiIGZvbnQtZmFtaWx5PSJ
zeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZvbnQtd2VpZ2h0PSJib2xkIiBm
aWxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0iZW5kIj7imqDvuI8gTk9URTwvdGV4dD4KICA8dGV4dCB4PSI3MzUiIHk9IjE3MiIgZm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC
1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSwwLjkpIiB0ZXh0LWFuY2hvcj0ibWlkZGx
lIj5wcm9jZXNzaW5nX3RpbWVfbXMgPSBOVUxMPC90ZXh0PgoKICA8IS0tIEFycm93IHRvIFN0ZXAg
MiAtLT4KICA8bGluZSB4MT0iNDI1IiB5MT0iMTg1IiB4Mj0iNDI1IiB5Mj0iMjEwIiBzdHJva2U9I
iM2NDc0OGIiIHN0cm9rZS13aWR0aD0iMiIgbWFya2VyLWVuZD0idXJsKCNwQXJyb3cpIi8+CgogID
whLS0gU3RlcCAyOiBPcHRpbWl6ZSBQcm9jZXNzaW5nIC0tPgogIDxyZWN0IHg9IjE1IiB5PSIyNSIgeT0iMjE
1IiB3aWR0aD0iNDAwIiBoZWlnaHQ9IjQwIiByeD0iOCIgZmlsbD0idXJsKCNwU3RlcEdyYWQpIiBm
aWx0ZXI9InVybCgjcFNoYWRvdykiLz4KICA8Y2lyY2xlIGN4PSIyNTUiIGN5PSIyMzUiIHI9IjE1
IiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuMikiLz4KICA8dGV4dCB4PSIyNTUiIHk9IjI0MSIgZm
9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxNCIgZm9udC13ZWl
naHQ9ImJvbGQiIGZpbGw9IiNmZmYiIHRleHQtYW5jaG9yPSJtaWRkbGUiPjI8L3RleHQ+CiAgPHRl
eHQgeD0iNDUwIiB5PSIyNDAiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvb
nQtc2l6ZT0iMTQiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0ibW
lkZGxlIj5PcHRpbWl6ZSBiYXNlZCBvbiBpc3N1ZSB0eXBlPC90ZXh0PgoKICA8IS0tIEFycm93IH
yYW5jaGVzIGZyb20gU3RlcCAyIC0tPgogIDxsaW5lIHgxPSIzMDAiIHkxPSIyNTUiIHgyPSIzMDAiIH
Vm9sdW1lIElzc3VlcyAtLT4KICA8cmVjdCB4PSIxMCIgeT0iMjcwIiB3aWR0aD0iMjYwIiBoZWlna
HQ9IjE4NSIgcng9IjgiIGZpbGw9InVybCgjcENhdGVnb3J5R3JhZCkiIGZpbHRlcj0idXJsKCNwU2
hhZG93KSIvPgogIDx0ZXh0IHg9IjE0MCIgeT0iMzE1IiBmb250LWZhbWlseT0ic3lzdGVtLXVpLCB
zYW5zLXNlcmlmIiBmb250LXNpemU9IjEzIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0iI2ZmZiIg
dGV4dC1hbmNob3I9Im1pZGRsZSI+SGlnaCBWb2x1bWUgSXNzdWVzPC90ZXh0PgoKICA8cmVjdCB4P
SI1NSIgeT0iMzMwIiB3aWR0aD0iMjMwIiBoZWlnaHQ9IjEzMCIgcng9IjYiIGZpbGw9InVybCgjcF
U1LDI1NSwyNTUsMC4xKSIvPgogIDx0ZXh0IHg9IjY0IiBnOTIHg9IjcwIiB3aWR0aD0iNTAiIGZvbnQtZmFtaWx5PSJzeXN
0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxs
PSIjYTVmM2ZjIj5Tb2x1dGlvbnM6PC90ZXh0PgogIDx0ZXh0IHg9IjgiIHk9IjcwIiBmb250LWZhb
mFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiNlMmU4Zj
AiPuKAoiBEcm9wIERFQlVHL2hlYWx0aCBjaGVja3MgZWFybHk8L3RleHQ+CiAgPHRleHQgeD0iNzA
iIHk9IjM4NyIgZm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIx
MCIgZmlsbD0iI2UyZThmMCI+4oCiIFJlZHVjZSBwYXJzaW5nIGNvbXBsZXhpdHk8L3RleHQ+CiAgP
HRleHQgeD0iNzAiIHk9IjQwNCIgZm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm
9udC1zaXplPSIxMCIgZmlsbD0iI2UyZThmMCI+4oCiIEF2b2lkIGV4cGVuc2l2ZSByZXBsYWNlQWx
sPC90ZXh0PgogIDx0ZXh0IHg9IjAiIHk9IjE4NyIgZm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iI2UyZThmMCI+4oCiIFVzZSBwYXRoIGZvbnQtZmFtaWx5PSJzeXN
hbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiNlMmU4ZjAiPuKAoiBVc2Ugc2FtcGxpbmcgZ
```

```
m9yIGhpZ2gtdm9sdW1lPC90ZXh0PgogIDx0ZXh0IHg9IjcwIiB5PSI0NDgiIGZvbnQtZmFtaWx5PS
JzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiM4NmVmYWMiPkV4cGV
jdGVkOiA2MC05MCUgcmVkdWN0aW9uPC90ZXh0PgoKICA8IS0tIENvbm5lY3QgdG8gSGlnaCBWb2x1
bWUgLS0+CiAgPGxpbmUgeDE9IjMwMCIgeTE9IjI4NSIgeDI9IjE3MCIgeTI9IjI4NSIgc3Ryb2tlP
SIjNjQ3NDhiIiBzdHJva2Utd2lkdGg9IjIiLz4KICA8bGluZSB4MT0iMTcwIiB5MT0iMjg1IiB4Mj
0iMTcwIiB5Mj0iMjkwIiBzdHJva2U9IiM2NDc0OGIiIHN0cm9rZS13aWR0aD0iMiIgbWFya2VyLWV
uZD0idXJsKCNwQXJyb3cpIi8+CgogIDwhLS0gQ2F0ZWdvcnkgMjogQ2FyZGluYWxpdHkgRXhwbG9z
aW9uIC0tPgogIDxyZWN0IHg9IjMxNSIgeT0iMjkwIiB3aWR0aD0iMjYwIiBoZWlnaHQ9IjE4NSIgc
ng9IjgiIGZpbGw9InVybCgjcENhdGVnb3J5R3JhZCkiIGZpbHRlcj0idXJsKCNwU2hhZG93KSIvPg
ogIDx0ZXh0IHg9IjQ0NSIgeT0iMzE1IiBmb250LWZhbWlseT0ic3lzdGVtLXVpLCBzYW5zLXNlcml
mIiBmb250LXNpemU9IjEzIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0iI2ZmZiIgdGV4dC1hbmNo
b3I9Im1pZGRsZSI+Q2FyZGluYWxpdHkgRXhwbG9zaW9uPC90ZXh0PgogICA8cmVjdCB4PSIzMCIg
Hk9IjMzMCIgd2lkdGg9IjIzMCIgaGVpZ2h0PSIxMzAiIHJ4PSI2IiBmaWxsPSJyZ2JhKDI1NSwyNT
UsMjU1LDAuMSkiLz4KICA8dGV4dCB4PSIzNDUiIHk9IjM1MCIgZm9udC1mYW1pbHk9InN5c3RlbS1
1aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IiNm
Y2QzNGQiPlN5bXB0b21zOjwvdGV4dD4KICA8dGV4dCB4PSIzNDUiIHk9IjM2NyIgZm9udC1mYW1pb
Hk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iI2UyZThmMCI+4o
CiIE1pbGxpb25zIG9mIHRpbWUgc2VyaWVzPC90ZXh0PgogIDx0ZXh0IHg9IjMzg0IiB
mb250LWZhbWlseT0ic3lzdGVtLXVpLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIj
ZTJlOGYwIj7igKIgSGlnaCBERFUgY29zdHM8L3RleHQ+CiAgPHRleHQgeD0iMzciIHk9IjM0MDUiI
GZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdd2
VpZ2h0PSJib2xkIiBmaWxsPSIjYTVmM2ZjIj5Tb2x1dGlvbnM6PC90ZXh0PgogIDx0ZXh0IHg9IjM
0NSIgeT0iNDIyIiBmb250LWZhbWlseT0ic3lzdGVtLXVpLCBzYW5zLXNlcmlmIiBmb250LXNpemU9
IjEwIiBmaWxsPSIjZTJlOGYwIj7igKIgQnVja2V0IGhpZ2gtY2FyZGluYWxpdHkgZmllbGRzPC90Z
Xh0PgogIDx0ZXh0IHg9IjM0NSIgeT0iNDM5IiBmb250LWZhbWlseT0ic3lzdGVtLXVpLCBzYW5zLX
NlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjZTJlOGYwIj7igKIgUmVtb3ZlIHVzZXJfaWQgZGl
tZW5zaW9uczwvdGV4dD4KICA8dGV4dCB4PSIzNDUiIHk9IjQ1NiIgZm9udC1mYW1pbHk9InN5c3Rl
bS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzg2ZWZhYyI+VGFyZ2V0OiAmb
HQ7MTBLIHNlcmllcy9tZXRyaWM8L3RleHQ+CgogIDwhLS0gQ29ubmVjdCB0byBDYXJkaW5hbGl0eS
AtLT4KICA8bGluZSB4MT0iNDI1IiB5MT0iMjg1IiB4Mj0iNDI1IiB5Mj0iMjkwIiBzdHJva2U9IiM
2NDc0OGIiIHN0cm9rZS13aWR0aD0iMiIgbWFya2VyLWVuZD0idXJsKCNwQXJyb3cpIi8+CgogIDwh
LS0gQ2F0ZWdvcnkgMzogQ29zdCBPcHRpbWl6YXRpb24gLS0+CiAgPHJlY3QgeD0iMzAiIHk9IjMy
TAiIHdpZHRoPSIyNDAiIGhlaWdodD0iMTg1IiByeD0iOCIgZmlsbD0iIzFlMjkzYiIgc3Ryb2tlP
FkKSIgZmlsdGVyPSJ1cmwoI3BTaGFkb3cpIi8+CiAgPHRleHQgeD0iMTUwIiB5PSIzMTUiIGZvbnQ
tZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdd2VpZ2h0
PSJib2xkIiBmaWxsPSIjZmZmIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5Db3N0IE9wdGltaXphdGlvb
jwvdGV4dD4KICA8dGV4dCB4PSIxNDAiIGhlaWdodD0iMTg1IiByeD0iOCIgZmlsbD0iIHdpZHRoPS
TAiIHdpZHRoPSIyNDAiIGhlaWdodD0iMTg1IiByeD0iOCIgZmlsbD0iIHdpZHRoPSIyNDAiIGhlaW
ByeD0iNiIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSwwLjEpIi8+CiAgPHRleHQgeD0iMTQwIiB5PSI
zNTAiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtZmFtaWx5PSJzeXN0
bnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjYTVmM2ZjIj5TdHJhdGVnaWVzOjwvdGV4dD4KICA8dGV4d
CB4PSIzMCIgeT0iMzMCIgZm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC
1zaXplPSIxMCIgZmlsbD0iI2UyZThmMCI+4oCiIERyb3AgREVCVUcvVFJBQ0UgbG9nczwvdGV4dD4
KICA8dGV4dCB4PSIzMCIgeT0iMzUwIiBmb250LWZhbWlseT0ic3lzdGVtLXVpLCBzYW5zLXNlcmlm
IiBmb250LXNpemU9IjEwIiBmaWxsPSIjZTJlOGYwIj7igKIgRXh0cmFjdCBtZXRyaWNzIGluc3RlY
WQ8L3RleHQ+CiAgPHRleHQgeD0iMzAiIHk9IjM4NyIgZm9udC1mYW1pbHk9InN5c3RlbS11aSwgc2
Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iI2UyZThmMCI+4oCiIFJlZHVjZSBkZXYgcmV
0ZW50aW9uPC90ZXh0PgogIDx0ZXh0IHg9IjE0MCIgeT0iNDA0IiBmb250LWZhbWlseT0ic3lzdGVtLXVpLCBzYW
5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjZTJlOGYwIj7igKIgRXh0cmFjdCBtZXRyaWN
zIGluc3RlYWQ8L3RleHQ+CiAgPHRleHQgeD0iMzAiIHk9IjQzNyIgZm9udC1mYW1pbHk9InN5c3Rl
ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtZmFtaWx5PSJzeXN0
```

SIjZmNkMzRkIj5ST0kgRXhhbXBsZTo8L3RleHQ+CiAgPHRleHQgeD0iNjIwIiB5PSI0NDgiIGZvbn
QtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiNlMmU
4ZjAiPjEwMEdCIGxvZ3Mg4oaSIE1ldHJpY3M8L3RleHQ+CiAgPHRleHQgeD0iNjIwIiB5PSI0NjIi
IGZvbnQtZmFtaWx5PSJzeXN0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9I
iM4NmVmYWMiPj0gOTkuOSUgY29zdCBzYXZpbmdzPC90ZXh0PgoKICA8IS0tIENvbm5lY3QgdG8gQ2
9zdCAtLT4KICA8bGluZSB4MT0iNTUwIiB5MT0iMjg1IiB4Mj0iNzEwIiB5Mj0iMjg1IiBzdHJva2U
9IiM2NDc0OGIiIHN0cm9rZS13aWR0aD0iMiIvPgogIDxsaW5lIHgxPSI3MTAiIHkxPSIyODUiIHgy
PSI3MTAiIHkyPSIyOTAiIHN0cm9rZT0iIzY0NzQ4YiIgc3Ryb2tlLXdpZHRoPSIyIiBtYXJrZXItZ
W5kPSJ1cmwoI3BBcnJvdykiLz4KCiAgPCEtLSBPcHRpbWl6YXRpb24gRXhhbXBsZXMgU2VjdGlvbi
AtLT4KICA8cmVjdCB4PSI0MCIgeT0iNDk1IiB3aWR0aD0iNzcwIiBoZWlnaHQ9IjE4NSIgcng9IjE
wIiBmaWxsPSIjMWUyOTNiIiBzdHJva2U9IiMzMzQxNTUiIHN0cm9rZS13aWR0aD0iMSIvPgogIDx0
ZXh0IHg9IjQyNSIgeT0iNTIwIiBmb250LWZhbWlseT0ic3lzdGVtLXVpLCBzYW5zLXNlcmlmIiBmb
250LXNpemU9IjE0IiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0iI2YxZjVmOSIgdGV4dC1hbmNob3
I9Im1pZGRsZSI+T3B0aW1pemF0aW9uIFBhdHRlcm5zPC90ZXh0PgoKICA8IS0tIFBhdHRlcm4gMTo
gRmlsdGVyIEZpcnN0IC0tPgogIDxyZWN0IHg9IjYwIiB5PSI1MzUiIHdpZHRoPSIyMzAiIGhlaWdo
dD0iMTMwIiByeD0iNiIgZmlsbD0iIzMzNDE1NSIvPgogIDx0ZXh0IHg9IjE3NSIgeT0iNTU1IiBmb
250LWZhbWlseT0ic3lzdGVtLXVpLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEyIiBmb250LXdlaW
dodD0iYm9sZCIgZmlsbD0iIzEwYjk4MSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+4pyTIEZpbHRlciB
GaXJzdCwgVGhlbiBQYXJzZTwvdGV4dD4KICA8dGV4dCB4PSI3NSIgeT0iNTgwIiBmb250LWZhbWls
eT0ibW9ub3NwYWNlIiBmb250LXNpemU9IjEwIiBmaWxsPSIjOTRhM2I4Ij7inYwgU0xPVzo8L3Rle
HQ+CiAgPHRleHQgeD0iNzUiIHk9IjU5NSIgZm9udC1mYW1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaX
plPSIxMCIgZmlsbD0iI2ZjYTVhNSI+cGFyc2UgfCBmaWx0ZXIgRVJST1I8L3RleHQ+CiAgPHRleHQ
geD0iNzUiIHk9IjYyMCIgZm9udC1mYW1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXplPSIxMCIgZmls
bD0iIzk0YTNiOCI+4pyTIEZBU1Q6PC90ZXh0PgogIDx0ZXh0IHg9Ijc1IiB5PSI2MzUiIGZvbnQtZ
mFtaWx5PSJtb25vc3BhY2UiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiM4NmVmYWMiPmZpbHRlciBFUl
JPUiB8IHBhcnNlPC90ZXh0PgogIDx0ZXh0IHg9Ijc1IiB5PSI2NTUiIGZvbnQtZmFtaWx5PSJzeXN
0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiM2NDc0OGIiPlJlZHVjZXMg
cGFyc2UgbG9hZCA3MCUrPC90ZXh0PgoKICA8IS0tIFBhdHRlcm4gMjogQ29tYmluZSBSZWdleCAtL
T4KICA8cmVjdCB4PSIzMTAiIHk9IjUzNSIgd2lkdGg9IjIzMCIgaGVpZ2h0PSIxMzAiIHJ4PSI2Ii
BmaWxsPSIjMzM0MTU1Ii8+CiAgPHRleHQgeD0iNDI1IiB5PSI1NTUiIGZvbnQtZmFtaWx5PSJzeXN
0ZW0tdWksIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTIiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxs
PSIjMTBiOTgxIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj7inJMgQ29tYmluZSBSZWdleCBQYXR0ZXJuc
zwvdGV4dD4KICA8dGV4dCB4PSIzMjUiIHk9IjU4MCIgZm9udC1mYW1pbHk9Im1vbm9zcGFjZSIgZm
9udC1zaXplPSIxMCIgZmlsbD0iIzk0YTNiOCI+4p2MIFNMT1c6PC90ZXh0PgogIDx0ZXh0IHg9IjM
yNSIgeT0iNTk1IiBmb250LWZhbWlseT0ibW9ub3NwYWNlIiBmb250LXNpemU9IjEwIiBmaWxsPSIj
ZmNhNWE1Ij5yZXBsYWNlQWxsKHAxKSB8IHJlcGxhY2VBbGwocDIpPC90ZXh0PgogIDx0ZXh0IHg9I
jMyNSIgeT0iNjIwIiBmb250LWZhbWlseT0ibW9ub3NwYWNlIiBmb250LXNpemU9IjEwIiBmaWxsPS
IjOTRhM2I4Ij7inJMgRkFTVDo8L3RleHQ+CiAgPHRleHQgeD0iMzI1IiB5PSI2MzUiIGZvbnQtZmF
taWx5PSJtb25vc3BhY2UiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiM4NmVmYWMiPnJlcGxhY2VBbGwo
IihwMXxwMikiLCAiWCIpPC90ZXh0PgogIDx0ZXh0IHg9IjMyNSIgeT0iNjU1IiBmb250LWZhbWlse
T0ic3lzdGVtLXVpLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjNjQ3NDhiIj5TaW
5nbGUgcGFzcyA9IDUwLTgwJSBmYXN0ZXI8L3RleHQ+CgogIDwhLS0gUGF0dGVybiAzOiBCdWNrZXR
pbmcgLS0+CiAgPHJlY3QgeD0iNTYwIiB5PSI1MzUiIHdpZHRoPSIyMzAiIGhlaWdodD0iMTMwIiBy
eD0iNiIgZmlsbD0iIzMzNDE1NSIvPgogIDx0ZXh0IHg9IjY3NSIgeT0iNTU1IiBmb250LWZhbWlse
T0ic3lzdGVtLXVpLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEyIiBmb250LXdlaWdodD0iYm9sZC
IgZmlsbD0iIzEwYjk4MSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+4pyTIEJ1Y2tldCBIaWdoIENhcmR
pbmFsaXR5PC90ZXh0PgogIDx0ZXh0IHg9IjU3NSIgeT0iNTgwIiBmb250LWZhbWlseT0ibW9ub3Nw
YWNlIiBmb250LXNpemU9IjEwIiBmaWxsPSIjOTRhM2I4Ij7inYwgQkFEOjwvdGV4dD4KICA8dGV4d

```
CB4PSI1NzUiIHk9IjU5NSIgZm9udC1mYW1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXplPSIxMCIgZm
lsbD0iI2ZjYTVhNSI+ZHVyYXRpb25fbXMgKGluZmluaXRlKTwvdGV4dD4KICA8dGV4dCB4PSI1NzU
iIHk9IjYyMCIgZm9udC1mYW1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzk0
YTNiOCI+4pyTIEdPT0Q6PC90ZXh0PgogIDx0ZXh0IHg9IjU3NSIgeT0iNjM1IiBmb250LWZhbWlse
T0ibW9ub3NwYWNlIiBmb250LXNpemU9IjEwIiBmaWxsPSIjODZlZmFjIj5kdXJhdGlvbl9idWNrZX
QgKDQgdmFsdWVzKTwvdGV4dD4KICA8dGV4dCB4PSI1NzUiIHk9IjY1NSIgZm9udC1mYW1pbHk9InN
5c3RlbS11aSwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzY0NzQ4YiI+OTkuOTYl
IHNlcmllcyBvZWR1Y3Rpb248L3RleHQ+Cjwvc3ZnPg==)



---

---

## Pipeline Health Monitoring

Monitor the overall health of your OpenPipeline configuration.

---

## Emergency Rollback Procedures ⭐ EXPANDED

When OpenPipeline issues occur in production, follow these emergency
procedures.

### Scenario 1: Critical Logs Being Dropped

**Symptoms:**
- Production error logs missing
- Security events not visible
- Business-critical logs vanished

**Immediate Actions:**

1. **Disable Drop Processor (UI)**
```
   Settings → Log Monitoring → OpenPipeline
   → Select pipeline → Processors tab
   → Find drop processor → Toggle OFF
   → Save pipeline

   Effect: Immediate (within 1–2 minutes)
```

2. **Disable Drop Processor (API)**
```bash
   # Get current pipeline config
```
```

```
   curl -X GET
"https://{tenant}.live.dynatrace.com/api/v2/openpipeline/logs/pipelines/{pipe
lineId}" \
     -H "Authorization: Api-Token {token}"

   # Edit JSON: Set processor "enabled": false

   # Update pipeline
   curl -X PUT
"https://{tenant}.live.dynatrace.com/api/v2/openpipeline/logs/pipelines/{pipe
lineId}" \
     -H "Authorization: Api-Token {token}" \
     -H "Content-Type: application/json" \
     -d @updated-pipeline.json
```

3. **Verify Logs Reappear**
```dql
   fetch logs
   | filter log.source == "critical-service"
   | filter timestamp > now() - 5m
   | summarize count()
```

Post-Incident:
- Review drop condition logic
- Add safeguards (never drop ERROR/FATAL)
- Test in staging before re-enabling

Scenario 2: Parsing Breaking Log Ingestion

Symptoms:
- Logs delayed or not appearing
- Pipeline processing time spiking
- Parse errors in pipeline metrics

Immediate Actions:

1. **Disable Parse Processor**
```
   Settings → OpenPipeline → Select pipeline
   → Disable DQL parse processor
   → Save

   Result: Logs stored unparsed (content field only)
   Impact: Fields not extracted, but logs visible
```

```

2. **Check Processing Metrics**
```
   ⚠️ QUERY DISABLED: Fields return NULL in this tenant

   dt.openpipeline.processing_time_ms - EXISTS but returns NULL
   dt.openpipeline.status - EXISTS but returns NULL
```

   **Alternative Query:**
```dql
// Monitor pipeline log volume and sources instead
fetch logs
| filter dt.openpipeline.pipelines == "my-pipeline"
| summarize
    log_count = count(),
    sources = collectDistinct(dt.openpipeline.source)
| sort log_count desc
```

3. **Test Parse Pattern Offline**
```dql
   // Use makeTimeseries or static data to test pattern
   data record(
     content = "192.168.1.1 - GET /api/users HTTP/1.1 200 1024"
   )
   | parse content, """
       IPADDR:client_ip SPACE '-' SPACE
       LD:method SPACE LD:path SPACE LD:protocol SPACE
       INT:status_code SPACE INT:bytes
     """
   | fields client_ip, method, status_code
```

Post-Incident:
- Fix parse pattern
- Test with production log samples
- Re-enable processor

Scenario 3: Masking Failure (PII/PHI Exposed)

Symptoms:
- Credit card numbers visible in queries
- SSNs, patient IDs, or emails not redacted
- Compliance violation alerts
```

````
**CRITICAL - Immediate Actions:**

1. **STOP INGESTION (if actively leaking)**
```bash
   # Disable entire pipeline immediately
   curl -X PUT
"https://{tenant}.live.dynatrace.com/api/v2/openpipeline/logs/pipelines/{pipe
lineId}" \
     -H "Authorization: Api-Token {token}" \
     -d '{"enabled": false}'
```

2. **Identify Exposure Window**
```dql
   fetch logs
   | filter log.source == "payment-service"
   | filter matchesPhrase(content, "4111-1111-1111-1111") // Test pattern
   | summarize
       first_exposure = min(timestamp),
       last_exposure = max(timestamp),
       affected_count = count()
```

3. **Purge Sensitive Data (if supported)**
```
   ⚠️ IMPORTANT: Dynatrace does not support selective log deletion

   Options:
   1. Wait for retention period to expire
   2. Contact Dynatrace support for emergency data purge
   3. Revoke bucket access to prevent further exposure
   4. Document incident for compliance audit
```

4. **Revoke Bucket Access**
```
   Settings → Buckets → Select bucket
   → Access Control → Remove all users except admins
   → Save

   Effect: Prevents unauthorized viewing while investigating
```

5. **Fix Masking and Re-Enable**
```dql
   // Test masking pattern before deploying
   data record(
````
````

```
      content = "Payment processed: card=4111-1111-1111-1111, cvv=123"
   )
   | fieldsAdd content = replaceAll(content, "\\b\\d{4}[\\s-]?\\d{4}[\\s-]?
\\d{4}[\\s-]?\\d{4}\\b", "[PAN_REDACTED]")
   | fieldsAdd content = replaceAll(content, "cvv[=:]?\\s*\\d{3,4}", "cvv=
[REDACTED]")
   | fields content

   // Expected: "Payment processed: card=[PAN_REDACTED], cvv=[REDACTED]"
```

Post-Incident (MANDATORY):
- [ ] Document exposure timeline
- [ ] Notify compliance/security team
- [ ] File incident report (PCI-DSS, HIPAA, GDPR)
- [ ] Review all masking patterns
- [ ] Implement automated compliance checks
- [ ] Add pre-production masking tests

Scenario 4: Routing to Wrong Bucket

Symptoms:
- Production logs in dev bucket (7-day retention)
- Audit logs in default bucket (missing compliance)
- High-value logs being dropped prematurely

Immediate Actions:

1. **Verify Current Routing**
```dql
   fetch logs
   | filter log.source == "production-api"
   | summarize {count = count()}, by: {dt.system.bucket}
| sort count desc
```

2. **Create Temporary High-Retention Bucket**
```
   Settings → Buckets → Create Bucket
   Name: emergency_recovery_logs
   Retention: 90 days
   Access: Admins only
```

3. **Update Pipeline Storage**
```
```

```
   Pipeline → Storage Configuration
   Default Bucket: emergency_recovery_logs
   Save

   Effect: New logs routed to safe bucket immediately
```

4. **Fix Routing Rules**
```
   Pipeline → Routing Tab

   Rule 1 (HIGHEST PRIORITY):
     Condition: log.source == "production-api" AND loglevel == "ERROR"
     Target: prod_error_logs (90 days)

   Rule 2:
     Condition: log.source == "production-api"
     Target: prod_logs (35 days)

   Default:
     Target: default_logs (35 days)
```

Post-Incident:
- Review all routing rules for correctness
- Document expected bucket for each log source
- Add monitoring alerts for unexpected bucket usage

Scenario 5: Complete Pipeline Rollback

When to Use:
- Multiple critical issues
- Unknown root cause
- Need time to investigate safely

Full Rollback Procedure:

1. **Disable Entire Pipeline**
```bash
   curl -X PUT
"https://{tenant}.live.dynatrace.com/api/v2/openpipeline/logs/pipelines/{pipe
lineId}" \
     -H "Authorization: Api-Token {token}" \
     -H "Content-Type: application/json" \
     -d '{"enabled": false}'
```

````
2. **Restore Previous Version (if available)**
```bash
   # Restore from backup JSON
   curl -X PUT
"https://{tenant}.live.dynatrace.com/api/v2/openpipeline/logs/pipelines/{pipe
lineId}" \
     -H "Authorization: Api-Token {token}" \
     -H "Content-Type: application/json" \
     -d @pipeline-backup-YYYYMMDD.json
```

3. **Verify Rollback Success**
```dql
   fetch logs
   | filter timestamp > now() - 5m
   | summarize
       count(),
       pipelines = collectDistinct(dt.openpipeline.pipelines),
       buckets = collectDistinct(dt.system.bucket)
```

4. **Communicate Status**
```
   Template:

   Subject: OpenPipeline Rollback - [Pipeline Name]

   Timeline:
   - [HH:MM] Issue detected: [description]
   - [HH:MM] Rollback initiated
   - [HH:MM] Rollback complete

   Impact:
   - Logs affected: [time window]
   - Data loss: [yes/no]
   - Service affected: [list]

   Next Steps:
   - Root cause analysis in progress
   - Fix ETA: [estimate]
   - Testing in staging before re-deploy
```

---

### Emergency Rollback Checklist
````

````
**Before ANY pipeline changes:**
- [ ] Backup current pipeline configuration (JSON export)
- [ ] Test in staging environment first
- [ ] Document expected behavior
- [ ] Have rollback plan ready
- [ ] Notify team of change window

**During incident:**
- [ ] Identify affected time window
- [ ] Disable problematic processor/pipeline
- [ ] Verify logs flowing again
- [ ] Document all actions taken
- [ ] Communicate to stakeholders

**After rollback:**
- [ ] Root cause analysis
- [ ] Fix and test in staging
- [ ] Peer review changes
- [ ] Document lessons learned
- [ ] Update runbooks

---



```python
// Overview: Log volume by pipeline (last 24 hours)
fetch logs, from: now() - 24h
| summarize {log_count = count()}, by: {dt.openpipeline.pipelines}
| sort log_count desc
```

---

## Performance Troubleshooting ⭐ NEW

Diagnose and resolve performance issues in high-volume OpenPipeline
environments.

### Performance Issue 1: Slow Processing / High Latency

**Symptoms:**
- Logs delayed (ingestion_time - timestamp > 5 minutes)
- Pipeline processing time > 100ms
- Queries timing out or slow to return

**Diagnosis Queries:**

```dql
````
````

```
// 1. Identify slow pipelines
// ⚠️ Original query disabled: dt.openpipeline.processing_time_ms returns
NULL
//
// fetch logs
// | filter timestamp > now() - 1h
// | summarize
//     avg_processing = avg(dt.openpipeline.processing_time_ms),
//     p95_processing = percentile(dt.openpipeline.processing_time_ms, 95),
//     max_processing = max(dt.openpipeline.processing_time_ms),
//     log_count = count(),
//     by: {dt.openpipeline.pipelines}
// | sort avg_processing desc

// Alternative: Analyze pipeline volume to identify high-load pipelines
fetch logs
| filter timestamp > now() - 1h
| summarize
    log_count = count(),
    sources = countDistinct(dt.openpipeline.source),
    logs_per_minute = count() / 60,
    by: {dt.openpipeline.pipelines}
| sort log_count desc
```

```dql
// 2. Check ingestion lag
// ⚠️ QUERY DISABLED: dt.system.ingestion_time returns NULL in this tenant
//
// Original query (commented out):
// fetch logs
// | filter timestamp > now() - 15m
// | fieldsAdd lag_seconds = (dt.system.ingestion_time - timestamp) /
1000000000
// | summarize
//     avg_lag = avg(lag_seconds),
//     p95_lag = percentile(lag_seconds, 95),
//     max_lag = max(lag_seconds),
//     by: {log.source}
// | filter avg_lag > 60
// | sort avg_lag desc
//
// Note: Ingestion lag monitoring requires dt.system.ingestion_time field
// which is not populated in this tenant.
```

```dql
// 3. Find expensive processors
```

```
// ⚠️ QUERY DISABLED: Processor-level metrics not available
//
// dt.openpipeline.processor_time_ms - Not available in this tenant
// dt.openpipeline.processor_name - Not available in this tenant
//
// Original query (commented out):
// fetch logs
// | filter dt.openpipeline.pipelines == "my-slow-pipeline"
// | summarize
//     processor_time = avg(dt.openpipeline.processor_time_ms),
//     by: {dt.openpipeline.processor_name}
// | sort processor_time desc
//
// Note: Processor-level performance metrics require additional telemetry
// configuration. Monitor overall pipeline performance instead (see Query 1
alternative).
```

Common Causes & Solutions:

Cause	Solution	Expected Improvement
Complex regex in parse/mask	Simplify patterns, use anchors	50-80%
faster
Too many processors	Consolidate logic, remove unused	30-50% faster
High-volume debug logs	Add drop processor early	60-90% reduction
Expensive fieldsAdd logic	Pre-compute values, use lookup	40-70% faster
Large log messages (>10KB)	Truncate content field	30-50% faster

Optimization Example:

```dql
// ❌ SLOW: Multiple replaceAll in sequence
| fieldsAdd content = replaceAll(content, "pattern1", "X")
| fieldsAdd content = replaceAll(content, "pattern2", "Y")
| fieldsAdd content = replaceAll(content, "pattern3", "Z")

// ✅ FAST: Single regex with alternation
| fieldsAdd content = replaceAll(content, "(pattern1|pattern2|pattern3)", "
[REDACTED]")
```

```dql
// ❌ SLOW: Parse all logs, then filter
parse content, """complex pattern"""
| filter loglevel == "ERROR"
```

```
// ✅ FAST: Filter first, then parse
| filter matchesPhrase(content, "ERROR")
| parse content, """complex pattern"""
```

Performance Issue 2: High Volume / Rate Limiting

Symptoms:
- HTTP 429 (Too Many Requests) errors
- Logs missing during peak traffic
- OneAgent warnings about rate limits

Diagnosis Queries:

```dql
// 1. Identify high-volume sources
// ⚠️ PARTIAL: dt.system.log_size_bytes returns NULL in this tenant
//
// Alternative: Use log count instead of byte size
fetch logs
| filter timestamp > now() - 1h
| summarize
    log_count = count(),
    logs_per_second = count() / 3600,
    by: {log.source}
| sort log_count desc
| limit 20
```

```dql
// 2. Check for volume spikes
// ⚠️ PARTIAL: dt.system.log_size_bytes returns NULL in this tenant
//
// Alternative: Monitor log count spikes instead
fetch logs
| filter timestamp > now() - 6h
| makeTimeseries
    log_count = count(),
    by: {log.source},
    interval: 5m
```

```dql
// 3. Analyze droppable logs
fetch logs
| filter timestamp > now() - 1h
```

```
| summarize
    total = count(),
    debug = countIf(loglevel == "DEBUG"),
    trace = countIf(loglevel == "TRACE"),
    health = countIf(contains(content, "/health")),
    by: {log.source}
| fieldsAdd droppable_pct = (debug + trace + health) / total * 100
| filter droppable_pct > 30
| sort droppable_pct desc
```

Volume Reduction Strategies:

Strategy	Use Case	Expected Reduction
Drop DEBUG/TRACE	Development, verbose apps	40-70%
Drop health checks	Microservices, K8s	20-40%
Sampling	High-volume, low-value	80-95% (with 10-20% sample)
Metric extraction	SLI metrics, counts	99%+ (logs → metrics)
Aggregation	Repeated messages	50-80%

Example: Aggressive Volume Reduction

```dql
// Drop low-value logs (in pipeline drop processor)
loglevel == "DEBUG" OR
loglevel == "TRACE" OR
contains(content, "/health") OR
contains(content, "/metrics") OR
matchesPhrase(content, "/ping") OR
(loglevel == "INFO" AND matchesPhrase(log.source, "chatty-service"))

// Expected: 60-80% volume reduction
```

Sampling Strategy:

```dql
// Sample 10% of INFO logs, keep all ERROR/WARN
| fieldsAdd sample_hash = hashMd5(toString(timestamp))
| filter
    loglevel == "ERROR" OR
    loglevel == "WARN" OR
    (loglevel == "INFO" AND sample_hash % 10 == 0)

// Result: ~70% reduction (if 80% are INFO)
```

````
---

### Performance Issue 3: Cardinality Explosion

**Symptoms:**
- Metric extraction creating millions of time series
- High DDU costs for metric storage
- Queries slow due to high cardinality

**Diagnosis Queries:**

```dql
// 1. Count unique dimension combinations
fetch logs
| filter timestamp > now() - 1h
| filter isNotNull(user_id)  // Example high-cardinality field
| summarize
    unique_users = countDistinct(user_id),
    unique_services = countDistinct(service.name),
    potential_series = countDistinct(concat(user_id, "_", service.name))
```

```dql
// 2. Estimate metric cardinality
fetch logs
| filter timestamp > now() - 24h
| summarize
    dim1_count = countDistinct(dimension1),
    dim2_count = countDistinct(dimension2),
    dim3_count = countDistinct(dimension3)
| fieldsAdd estimated_series = dim1_count * dim2_count * dim3_count

// ⚠️ WARNING: If estimated_series > 100,000 → cardinality problem!
```

```dql
// 3. Find high-cardinality fields
fetch logs
| filter timestamp > now() - 1h
| summarize
    user_ids = countDistinct(user_id),
    request_ids = countDistinct(request_id),
    session_ids = countDistinct(session_id),
    paths = countDistinct(request_path),
    by: {service.name}
| sort user_ids desc
```
````

````
**Cardinality Reduction Techniques:**

| Problem | ❌ High Cardinality | ✅ Low Cardinality | Reduction |
|---------|-------------------|------------------|----------|
| User tracking | `user_id` (1M unique) | `user_segment` (5 values) |
99.9995% |
| API paths | `/api/users/12345` | `/api/users/{id}` | 99% |
| Durations | `duration_ms` (infinite) | `duration_bucket` (10 buckets) |
99.9%+ |
| Timestamps | `timestamp` (infinite) | `hour_of_day` (24 values) | 99.9%+ |
| IP addresses | `client_ip` (100K unique) | `client_country` (200 values) |
99.8% |

**Example: Bucketing Strategy**

```dql
// ❌ BAD: Infinite cardinality
| fieldsAdd duration_ms = response_time
// Metric extraction: log.api.response_time (dimensions: service, path,
duration_ms)
// Result: 5 services × 100 paths × 10,000 durations = 5,000,000 time series
❌

// ✅ GOOD: Bucketed cardinality
| fieldsAdd duration_bucket =
    if(response_time < 100, "fast",
    else: if(response_time < 500, "normal",
    else: if(response_time < 2000, "slow",
    else: "very_slow")))
// Metric extraction: log.api.response_time (dimensions: service, path,
duration_bucket)
// Result: 5 services × 100 paths × 4 buckets = 2,000 time series ✅
// Reduction: 99.96%
```

**Example: Path Normalization**

```dql
// ❌ BAD: /api/users/12345, /api/users/67890, ... (infinite paths)
| fieldsAdd api_path = request_path

// ✅ GOOD: Normalize to /api/users/{id}
| fieldsAdd api_path = replaceAll(request_path, "/api/users/\\d+",
"/api/users/{id}")
| fieldsAdd api_path = replaceAll(api_path, "/api/orders/\\d+",
"/api/orders/{id}")
| fieldsAdd api_path = replaceAll(api_path, "/api/products/[a-f0-9-]+",
"/api/products/{uuid}")
````
````

```
// Result: 20 normalized paths instead of 100,000+ unique paths
```

Cardinality Budget:

```
Target: < 10,000 time series per metric

Good Examples:
- 5 services × 20 endpoints × 3 status_categories = 300 series ✅
- 10 services × 5 environments × 4 duration_buckets = 200 series ✅

Bad Examples:
- 5 services × 100K user_ids = 500,000 series ❌
- 10 services × 10K request_ids × 5 methods = 500,000 series ❌

Rule of Thumb:
If dimension has > 1,000 unique values → BUCKET IT or REMOVE IT
```

```python
// Pipeline volume trend over time
fetch logs, from: now() - 24h
| makeTimeseries {log_count = count()}, by: {dt.openpipeline.pipelines},
interval: 1h
```

```python
// Check for logs without pipeline assignment (potential routing issues)
fetch logs, from: now() - 1h
| filter isNull(dt.openpipeline.pipelines)
| summarize {unrouted_count = count()}, by: {log.source}
| sort unrouted_count desc
```

```python
// Bucket distribution check
fetch logs, from: now() - 24h
| summarize {
    log_count = count()
  }, by: {dt.system.bucket, dt.openpipeline.pipelines}
| sort log_count desc
```

```python
```

```
// Log source health - are all expected sources sending data?
fetch logs, from: now() - 1h
| summarize {
    log_count = count(),
    last_seen = max(timestamp)
  }, by: {log.source}
| fieldsAdd minutes_since_last = (now() - last_seen) / 60000000000
| sort minutes_since_last desc
```

Common Issues & Solutions

Issue 1: Logs Not Appearing

Symptoms:
- Expected logs missing from queries
- Volume lower than expected

Diagnosis:

```python
// Check if logs are being dropped
// Compare volume before and after drop processors
fetch logs, from: now() - 1h
| summarize {
    total = count(),
    debug = countIf(loglevel == "DEBUG"),
    trace = countIf(loglevel == "TRACE")
  }, by: {log.source}
```

Solution:
1. Check drop processor conditions in pipeline configuration
2. Verify routing conditions aren't too restrictive
3. Confirm log source is sending data to Dynatrace
4. Check for OneAgent or log forwarder issues

Issue 2: Parsing Not Working

Symptoms:
- `loglevel` field is null
- Expected parsed fields missing

Diagnosis:

```python
```

```
// Find logs with parsing failures
fetch logs, from: now() - 1h
| filter isNull(loglevel)
| fields timestamp, log.source, dt.openpipeline.pipelines, content
| limit 20
```

Solution:
1. Test DPL pattern in DPL Architect with failing log samples
2. Check for variations in log format
3. Verify processor matching condition includes these logs
4. Order processors correctly (parse before fieldsAdd that uses parsed fields)

Issue 3: Wrong Pipeline Assignment

Symptoms:
- Logs in unexpected pipelines
- Routing not working as expected

Diagnosis:

```python
// Check routing - are logs going to expected pipelines?
fetch logs, from: now() - 1h
| summarize {log_count = count()}, by: {log.source,
dt.openpipeline.pipelines}
| sort log.source asc
```

Solution:
1. Review dynamic routing table configuration
2. Check routing condition order (first match wins)
3. Verify matching conditions use correct fields
4. Test conditions with sample logs

Issue 4: Masking Not Applied

Symptoms:
- Sensitive data visible in logs
- Redaction placeholders missing

Diagnosis:

```python
// Check if masking is working
fetch logs, from: now() - 1h
| filter contains(content, "@")  // Potential email
```

```
| filter NOT contains(content, "[EMAIL_REDACTED]")
| fields content
| limit 20
```

Solution:
1. Verify masking processor is in the pipeline
2. Check masking processor order (should be first)
3. Test DPL pattern matches the sensitive data format
4. Ensure masking applies to correct fields

Issue 5: Metrics Not Generated

Symptoms:
- Expected metrics missing
- Metric queries return no data

Diagnosis:

```python
// List log-extracted metrics
// Note: Use the Dynatrace UI (Observe > Metrics) to browse metrics
// Or use timeseries to query a specific metric:
// timeseries avg_value = avg(log.your_metric_name), from: now() - 24h
```

Solution:
1. Verify extraction processor configuration
2. Check matching condition selects correct logs
3. Ensure value field exists and is numeric (for value metrics)
4. Check for dimension cardinality issues

Parsing Validation

Comprehensive checks for parsing effectiveness.

```python
// Parsing success rate by pipeline
fetch logs, from: now() - 24h
| summarize {
    total = count(),
    with_loglevel = countIf(isNotNull(loglevel)),
    without_loglevel = countIf(isNull(loglevel))
  }, by: {dt.openpipeline.pipelines}
| fieldsAdd parse_rate = round((toDouble(with_loglevel) / toDouble(total)) *
100, decimals: 1)
```

```
| sort parse_rate asc
```

```python
// Log level distribution (verify expected values)
fetch logs, from: now() - 24h
| summarize {log_count = count()}, by: {loglevel}
| sort log_count desc
```

```python
// Find log sources with low parsing rates
fetch logs, from: now() - 24h
| summarize {
    total = count(),
    parsed = countIf(isNotNull(loglevel))
  }, by: {log.source}
| fieldsAdd parse_rate = round((toDouble(parsed) / toDouble(total)) * 100,
decimals: 1)
| filter parse_rate < 90
| sort parse_rate asc
```

```python
// Sample unparsed logs from problematic sources
// Replace 'your-source' with source from previous query
fetch logs, from: now() - 1h
| filter log.source == "your-source"
| filter isNull(loglevel)
| fields content
| limit 30
```

```python
// Verify custom field extraction
// Add your expected custom fields here
fetch logs, from: now() - 24h
| summarize {
    total = count(),
    with_request_id = countIf(isNotNull(request_id)),
    with_user_id = countIf(isNotNull(user_id)),
    with_order_id = countIf(isNotNull(order_id))
  }, by: {dt.openpipeline.pipelines}
```

Volume & Cost Validation

Verify data volumes and cost optimization effectiveness.

```python
// Daily log volume by bucket (cost impact)
fetch logs, from: now() - 7d
| fieldsAdd day = formatTimestamp(timestamp, format: "yyyy-MM-dd")
| summarize {log_count = count()}, by: {day, dt.system.bucket}
| sort day desc, log_count desc
```

```python
// Volume trend - verify stable after migration
fetch logs, from: now() - 7d
| makeTimeseries {log_count = count()}, interval: 1h
```

```python
// Check if debug/trace logs are being dropped (cost savings)
fetch logs, from: now() - 24h
| filter loglevel == "DEBUG" OR loglevel == "TRACE"
| summarize {debug_trace_count = count()}
| fieldsAdd status = if(debug_trace_count == 0,
    "✅ Debug/trace logs dropped as expected",
    else: "⚠️ Debug/trace logs still present - check drop processors")
```

```python
// Volume by log level (should see mostly INFO/WARN/ERROR)
fetch logs, from: now() - 24h
| summarize {log_count = count()}, by: {loglevel}
| fieldsAdd percentage = round((toDouble(log_count) / 100), decimals: 1)
| sort log_count desc
```

```python
// Top 10 highest volume sources
fetch logs, from: now() - 24h
| summarize {log_count = count()}, by: {log.source}
| sort log_count desc
| limit 10
```

Performance Optimization

Tips for optimizing OpenPipeline performance.

````
### Processor Ordering

1. **Masking** - Always first (security)
2. **Drop** - Second (reduce volume early)
3. **Technology Parsers** - Before custom parsing
4. **Custom Parse** - Before fieldsAdd that uses parsed fields
5. **fieldsAdd** - After parsing, for computed fields
6. **fieldsRemove** - Last, to clean up temporary fields

### Matching Condition Optimization

| Do | Don't |
|----|-------|
| Use indexed fields | Complex regex on content |
| Check field existence first | Expensive operations always |
| Simple equality checks | Multiple OR conditions |

### Pattern Efficiency

| Efficient | Less Efficient |
|-----------|----------------|
| `'prefix=' LD:val` | `LD 'prefix=' LD:val` |
| Specific literals | Greedy matchers |
| Short patterns | Long complex patterns |

```python
// Identify pipelines with high volume (candidates for optimization)
fetch logs, from: now() - 1h
| summarize {hourly_volume = count()}, by: {dt.openpipeline.pipelines}
| sort hourly_volume desc
```

---

## Rollback Procedures

If issues are detected, here's how to rollback.

### Rollback Options

| Level | Action |
|-------|--------|
| **Pipeline** | Disable specific pipeline, use default |
| **Processor** | Disable individual processor |
| **Routing** | Reset dynamic routing to defaults |
| **Full** | Revert to classic ingestion (pre-v1.295) |
````

````
### Quick Disable Steps

1. Navigate to **Settings → OpenPipeline → Logs**
2. Select the problematic pipeline
3. Toggle the pipeline off
4. Logs will route to default pipeline

### Version History

OpenPipeline maintains configuration history:

1. Click on pipeline
2. View **History** tab
3. Select previous version
4. **Restore** to that version

---

## Ongoing Maintenance

After migration, establish these maintenance practices.

### Daily Checks

- [ ] Log volume within expected range
- [ ] No spike in unparsed logs
- [ ] Error log levels visible

### Weekly Checks

- [ ] Parsing success rates stable
- [ ] Bucket distribution as expected
- [ ] New log sources properly routed

### Monthly Checks

- [ ] Review masking effectiveness
- [ ] Optimize high-volume pipelines
- [ ] Clean up unused pipelines
- [ ] Document any changes

```python
// Weekly health report query
fetch logs, from: now() - 7d
| summarize {
    total_logs = count(),
    parsed_logs = countIf(isNotNull(loglevel)),
    error_logs = countIf(loglevel == "ERROR"),
````
````

````
    pipelines_used = countDistinct(dt.openpipeline.pipelines),
    sources_active = countDistinct(log.source)
  }
| fieldsAdd parse_rate = round((toDouble(parsed_logs) / toDouble(total_logs))
* 100, decimals: 1)
| fieldsAdd error_rate = round((toDouble(error_logs) / toDouble(total_logs))
* 100, decimals: 2)
```

```python
// Find new log sources that may need pipeline configuration
fetch logs, from: now() - 24h
| filter isNull(dt.openpipeline.pipelines) OR dt.openpipeline.pipelines ==
"default"
| summarize {
    log_count = count(),
    first_seen = min(timestamp)
  }, by: {log.source}
| filter log_count > 100
| sort log_count desc
```

---

## Migration Complete Checklist

Final validation before declaring migration complete.

### Data Quality ✅

- [ ] All log sources flowing
- [ ] Parsing rates > 95%
- [ ] Log levels correctly extracted
- [ ] Custom fields available
- [ ] Timestamps accurate

### Routing ✅

- [ ] All sources routed to correct pipelines
- [ ] Bucket assignments verified
- [ ] Dynamic routing working
- [ ] No orphaned data

### Security ✅

- [ ] All PII masked
- [ ] Compliance requirements met
- [ ] No sensitive data in stored logs
````
````

```
- [ ] Masking verified with queries

### Cost ✅

- [ ] Debug/trace logs dropped
- [ ] Volumes as expected
- [ ] Bucket retention configured
- [ ] Cost projections met

### Extraction ✅

- [ ] Metrics generating
- [ ] Events appearing
- [ ] Business events available
- [ ] Davis AI integration working

### Documentation ✅

- [ ] Pipeline configuration documented
- [ ] Routing rules documented
- [ ] Masking patterns documented
- [ ] Runbooks updated

### Team ✅

- [ ] Team trained on OpenPipeline
- [ ] Support procedures updated
- [ ] Escalation paths defined
- [ ] Monitoring dashboards created

```python
// Final migration validation summary
fetch logs, from: now() - 24h
| summarize {
    total_logs = count(),
    with_pipeline = countIf(isNotNull(dt.openpipeline.pipelines)),
    with_loglevel = countIf(isNotNull(loglevel)),
    with_masking = countIf(contains(content, "REDACTED")),
    unique_sources = countDistinct(log.source),
    unique_pipelines = countDistinct(dt.openpipeline.pipelines)
  }
| fieldsAdd
    routing_health = if(with_pipeline > total_logs * 0.95, "✅ Good", else:
"⚠️ Check routing"),
    parsing_health = if(with_loglevel > total_logs * 0.90, "✅ Good", else:
"⚠️ Check parsing")
```
```

```
---

## 🎉 Congratulations!

You've completed the OpenPipeline migration notebook series!

### What You've Learned

| Notebook | Topic |
|----------|-------|
| **OPMIG-01** | Introduction & Why Migrate |
| **OPMIG-02** | Architecture & Key Concepts |
| **OPMIG-03** | Migration Assessment & Planning |
| **OPMIG-04** | Pipeline Configuration |
| **OPMIG-05** | Routing & Buckets |
| **OPMIG-06** | Processing & Parsing |
| **OPMIG-07** | Metric & Event Extraction |
| **OPMIG-08** | Security & Masking |
| **OPMIG-09** | Troubleshooting & Validation |

### Next Steps

1. **Share knowledge** - Train your team
2. **Create dashboards** - Monitor pipeline health
3. **Iterate** - Add pipelines for new use cases
4. **Optimize** - Continuously improve performance

---

## References

- [OpenPipeline Documentation](https://docs.dynatrace.com/docs/discover-dynatrace/platform/openpipeline)
- [OpenPipeline Best Practices](https://docs.dynatrace.com/docs/discover-dynatrace/platform/openpipeline/best-practices)
- [DQL Reference](https://docs.dynatrace.com/docs/discover-dynatrace/platform/grail/dynatrace-query-language)
- [Dynatrace Community](https://community.dynatrace.com/)

---

*Last Updated: December 12, 2025*
```