1 FEBRUARY 2023

SUPPORTUTILS

Everything you need (most of the time) to resolve any SUSE Linux Enterprise technical issue.

Presentation by Grant Marcroft
Premium Services Engineer
(Dedicated) for Azure CSS

Skills required to “effectively” work on SUSE cases

1. Familiarity with using a console/terminal (emulator)
2. Familiarity with using pagers such as `less` (or `more`)
3. Basic Linux Administration skills
4. (optional) Be an open-source enthusiast

`courier` font indicates a command, a filename or a path
Liberation Sans font indicates a URL
Poppins font used for everything else

What Azure CSS can handle without an embedded SUSE Premium Engineer

The primary role of L1 in SUSE's internal support structure is information collection and basic analysis. Case Owners (COs) in CSS should be able to handle this role.

The role of L2 is more technical and in-depth data analysis, to validate CO's work and problem duplication. Linux Escalation Team members should be able to handle this role.

Technical issues not caused by a defect

(L1 → L2)

Root cause analysis of failures and general usage information about SUSE software is covered here.

If there is public documentation available or an Azure CSS team member has enough background in the technology in question, there is no need to involve SUSE Support.

Keep in mind, Pay-As-You-Go (PAYG) customers are not entitled to direct support from SUSE. If a customer wants a direct support relationship with SUSE, they should start with a Bring-Your-Own-Subscription (BYOS) image or migrate their existing PAYG workloads to BYOS with Azure Hybrid Benefit (AHB).

What Azure CSS cannot handle without an embedded SUSE Premium Engineer

Bugzilla tickets can be opened outside a SUSE Service Request; however, these are **not** "queued up" for Engineering until they're in our internal tool "Solid Ground." Some Microsoft Engineering contacts have bugzilla access for their roles but even they must open a support ticket with SUSE to follow the correct defect resolution workflow. A SUSE Service Request fast-tracks defect resolution.

Reporting bugs and resolving software defects

(L2 → L3)

SUSE has a standard "L3" process to triage defects reported by SUSE customers. The process always starts with a case opened with SUSE Support. If SUSE Support fully isolates a defect by duplicating the issue, they can submit it to bugzilla and the internal collaboration engineer (SME) enters this ticket into our "Solid Ground" system after validating the data. If Support cannot solve a incident even after reaching out to the Subject Matter Expert, they may submit the issue as an "L3 Question"

Bugzilla to SUSE == IcM to Microsoft

Unless it's a software defect, it's the Support teams' responsibility to fix it!

Microsoft CSS's staffing helps augment the need for SUSE Engineers in the Azure cloud. Software licenses for Pay-As-You-Go (PAYG) images are paid to Microsoft to provide this additional staffing and support to end-customers. This is the SUSE Cloud Service Provider (CSP) program and is the same for all major cloud providers.

Azure CSS's staffing numbers:

Thousands and available 24/7

Premium Engineers assigned to CSS:

Four, but only two dedicated to CSS. Available during their Geo's regular business hours

Andi in Germany (Central Euro Time)
Grant in Seattle (Pacific Time)

The two other engineers also support other SUSE customers

We work like a TAM and fill in any support gaps with the world's PAYG customers that can't be handled by CSS alone. Because of our extreme staffing ratio and global customer presence, it is crucial that SUSE Engineers' time is reserved for critical situations and/or work that can only be done via vendor collaboration.

(But you can always IM me if you have a quick question or want to say hi!)

Common Scenarios Outside Scope of Support

1. Consulting – Not a simple break/fix issue

Separate offering. Project-based work.

- System architecture issues
- Software configuration assistance
- Top-down system or cluster health checks
- Premium Technical Advisory Service (PTAS)

Consulting work from certified SUSE architects is competitively priced.

2. Administration

SUSE and Microsoft not responsible for end-customer's administration work. (e.g. Data/log collection, implementing solution based on CSS/SUSE feedback.)

3. Problems that involve third-party software or kernel modules

- This includes SAP Application problems and anti-virus/endpoint protection software.
- Compatibility problems with third-party software must be addressed by the third-party vendor's Engineering team`

4. Scheduled Standby

Separate offering. Purchased in four-hour blocks >2 weeks in advance.

SUSE Global Support staffs on weekends for Sev 1 (CritSit) issues only. Our organization has one primary and one backup engineer who volunteers to work "pager" duty in addition to their regular five day work week.

Azure's Premium Engineers are not expected to work weekends; only regular business hours, but they work until their work is finished. Hence, there is no one in SUSE support working a regular shift over the weekend to make sure an end-customer's off-hours maintenance work goes according to plan. Best for after-hours maintenance and operators carrying out go-live tasks.

So now, what is supportconfig?

- Collects *pretty much* everything one needs to diagnose an issue with SUSE base system software.

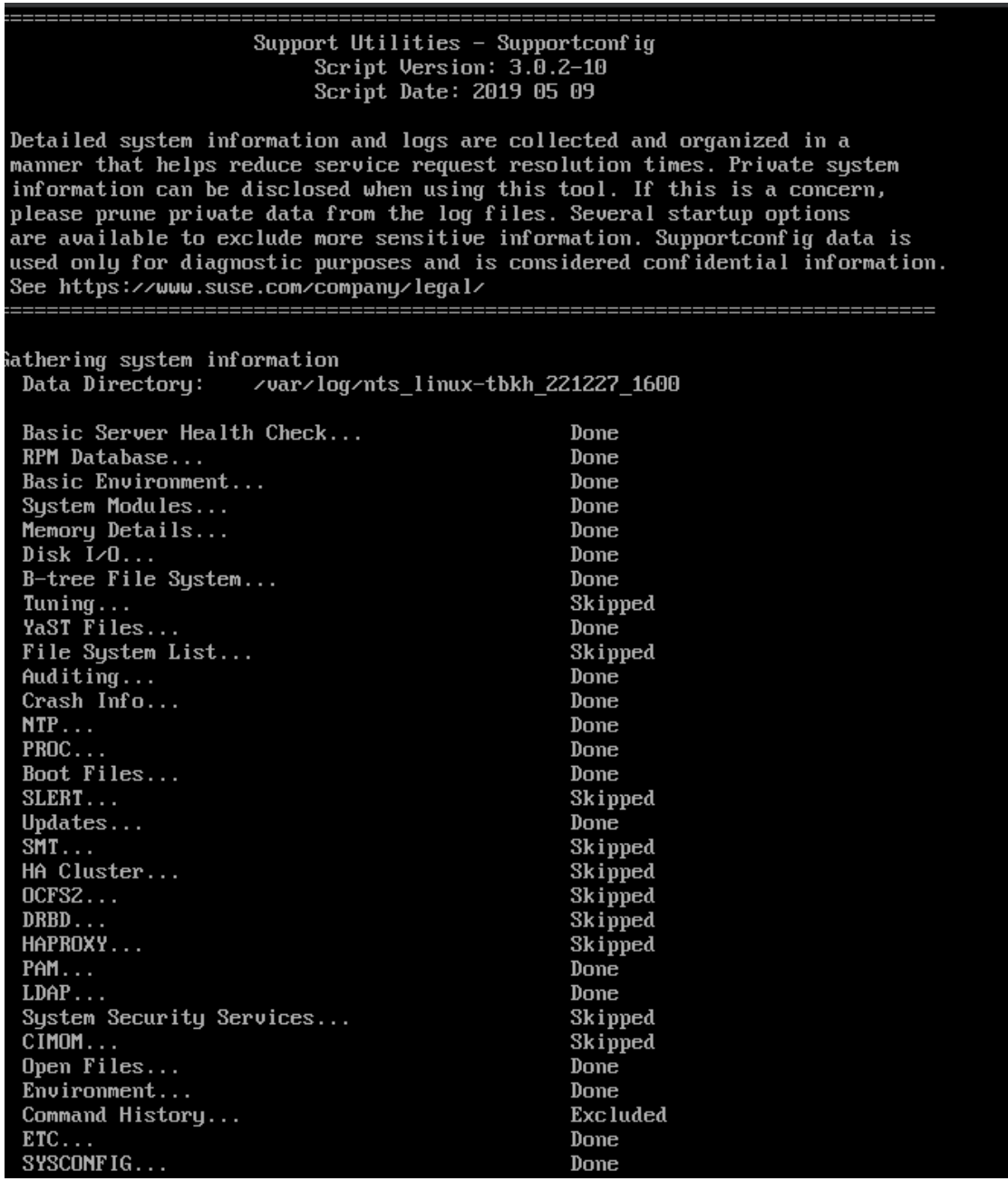
- Developed and primarily maintained by Jason Record, a 25+ year SUSE Support Veteran.

- Part of the “supportutils” package. (Installed by default)

- Filename format: org_hostname_date_time.txz

Old supportutils uses “nts_” prefix for Novell Technical Services. New supportutils updated to use “scc_” for SUSE Customer Center.

- Custom plugins can be developed. Public Cloud and SAP plugins already available.



First places to check

`basic-environment.txt` – Identify OS and Service Pack.
This info is crucial for any Engineering involvement.
Also contains `uname -a` (running kernel version) and `date` output.

SPs beyond General Support are best-effort only. No L3 support.
ESPOS with SAP images or LTSS (BYOS only) have extend lifecycles.

`basic-health-check.txt` – Check for adminstrator oversights and obvious server health issues

`updates.txt` – Check product info, patch status, issues with zypper and general repository info

`public_cloud/` directory – Packages and information specific to Public Cloud Update Infrastructure registration

`crash.txt` – Check whether system generated a kdump on unexpected reboot

Public Cloud plugin

Developed by Public Cloud Engineering team. Installed by default on cloud images.
Output files written to `public_cloud/` directory in `supportconfig` archive.

`cloudregister.txt` – Registration logs*
`credentials.txt` – Client credentials given to Public Cloud Update Repository servers
`frameworkpackages.txt` – Packages specific to providing the registration*
`framework.txt` – Should always say “Azure” in the Microsoft cloud
`hosts.txt` – Lists the RMT server IP on successful registration*
`instanceinit.txt` – `cloud-init` and `waagent` cloud instance init logs.
`metadata.txt` – Metadata pulled from Azure Instance Metadata Service (IMDS)*
`osrelease.txt` – Major OS and Service Pack version info
`regionserverclnt.cfg` – Client configuration file for PAYG registration*
`repositories.txt` – List of configured repos
`services.txt` – List of configured services
`updateinfrastructure.txt` – Output of “`zypper ref`” command

*: Most important files for diagnosing registration issues. Others are more or less informational

SAP HA Plugin

Developed by SAP Subject Matter Experts in SUSE Global Support

- Useful for basic checks on NetWeaver and HANA cluster nodes.

- Checks that needed RPMs are installed, detects running instances and displays information about them.

- Included by default on latest SLES for SAP images.

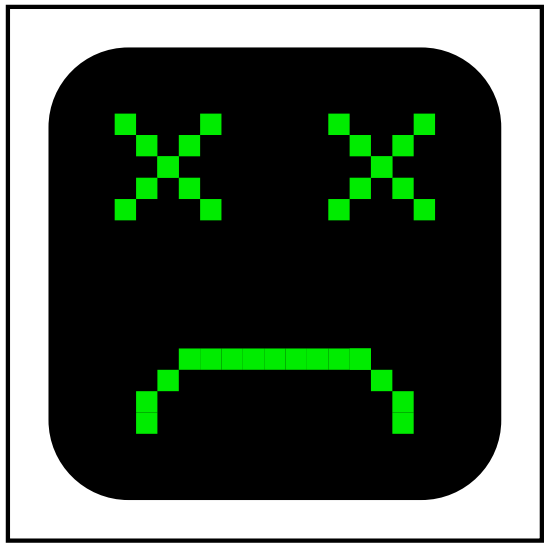
- Provides documentation URLs for SAP tuning, best-practices guides, etc.

- **NOTE**: SAP help from anyone other than SAP is best-effort support only. The SAP organization wants all SAP Application issues to go through their support channels.

Official TID:

SLES for SAP - How To Engage SAP and SUSE to address Product Issues
https://www.suse.com/support/kb/doc/?id=000019000`

System Crashes and Unexpected Reboots



- First place to check after an unexpected reboot:
`/var/crash` directory.

- Crash dump directories named with format:
`yyyy-mm-dd-hh:MM/`

- If a kdump is written, SUSE Support can investigate further.

- You can check basic info and backtrace of crash in
`yyyy-mm-dd-hh:MM/dmesg.txt` without installing any debugging tools.

- If crash directory contains only `dmesg.txt` and `vmcore`, dump may be corrupt/incomplete.
If there is a `readme.txt`, kdump's "`mkdumpfile`" should have completed successfully.

- Third-party kernel module defects are common cause of kernel panicks. If crash only occurs when third-party software in use, customer needs to contact the module vendor.

`analyzevmcore` tool

```
==============================================
      Support Utilities - Analyzevmcore
          Script Version: 1.38
          Script Date: 2021 05 25
==============================================

Analyzing 1 of 1 /var/crash/2023-01-03-15:08...
  Existing analysis...            not found
  Checking vmlinux...             validated (vmlinux-4.4.180-94.164-default.gz)
  Checking vmlinux.debug...       validated
  Creating analysis...            done
  Final analysis:                 /var/log/scc_analyzevmcore_2023-01-03-15:08.txt

Finished!

linux-tbkh:~ # _
```

- Generate a basic kdump report on a local system. Kdump analysis self-service.

- Instructs end user on missing debugging software dependencies to prepare for core dump analysis.

- Should only be used on customer system if they are willing and allowed to add debug software. Ideally, only in Dev/Test environments.

- Support teams can also upload a customer's core to their own test system and run it there. They may be able to disqualify the crash as an OS/Kernel defect without needing to engage SUSE Support.

- Writes report under "`/var/log/`". Automatically picked up by `supportconfig` and added to `crash.txt`

```
################################################################################
Binary Check Tool, v1.02.1
Date:   01/06/23, 15:39:58
Kernel: 4.4.180-94.164-default, Hardware: x86_64
################################################################################

Checking Binary Ownership                                   ... Done
Checking for Shared Libraries                               ... ERROR
Validating Unique RPMs                                      ... Done

################################################################################
Binary Checked: /usr/sbin/tcpdump
Log File:       /var/log/scc_chkbin_tcpdump_4625.txt
STATUS:         ERROR
################################################################################

linux-tbkh:/var/log # cat /var/log/scc_chkbin_tcpdump_4625.txt
################################################################################
Binary Check Tool, v1.02.1
Date:   01/06/23, 15:39:58
Kernel: 4.4.180-94.164-default, Hardware: x86_64
################################################################################

Checking Binary Ownership                                   ... Done
Checking for Shared Libraries                               ... ERROR
Validating Unique RPMs                                      ... Done

#--[ Checking File Ownership ]-------------------------------------#
/usr/sbin/tcpdump                  - from RPM: tcpdump-4.9.2-14.20.1.x86_64
 :ldd /usr/sbin/tcpdump                                [   ERROR     ]
/usr/sbin/tcpdump: error while loading shared libraries: /usr/lib64/libpcap.so.1: invalid ELF header

#--[ Validating Unique RPMs ]--------------------------------------#
Validating RPM: tcpdump-4.9.2-14.20.1.x86_64          [   Passed    ]


################################################################################
Binary Checked: /usr/sbin/tcpdump
Log File:       /var/log/scc_chkbin_tcpdump_4625.txt
STATUS:         ERROR
################################################################################

linux-tbkh:/var/log # _
```

`chkbin` tool

- Analogous to `zypper`'s role with package dependency resolution for `rpm` installations.

- Runs an "`rpm --verify`" on a executable binary **and** all of its library dependencies.

- Useful to find corrupted binaries and libraries when programs fail to execute/crash on startup

- Creates a report in "`/var/log/`" which is picked up by supportconfig. Added to `crash.txt`.

`getappcore` tool

- Prepare a user-space application core dump for analysis by SUSE Engineering.

- `crash.txt` file lists any cores written to the systemd journal.

- The binary and library dependencies of the failed program are copied into a archive in the "`/var/log`" directory. This can be provided to SUSE support for further debugging

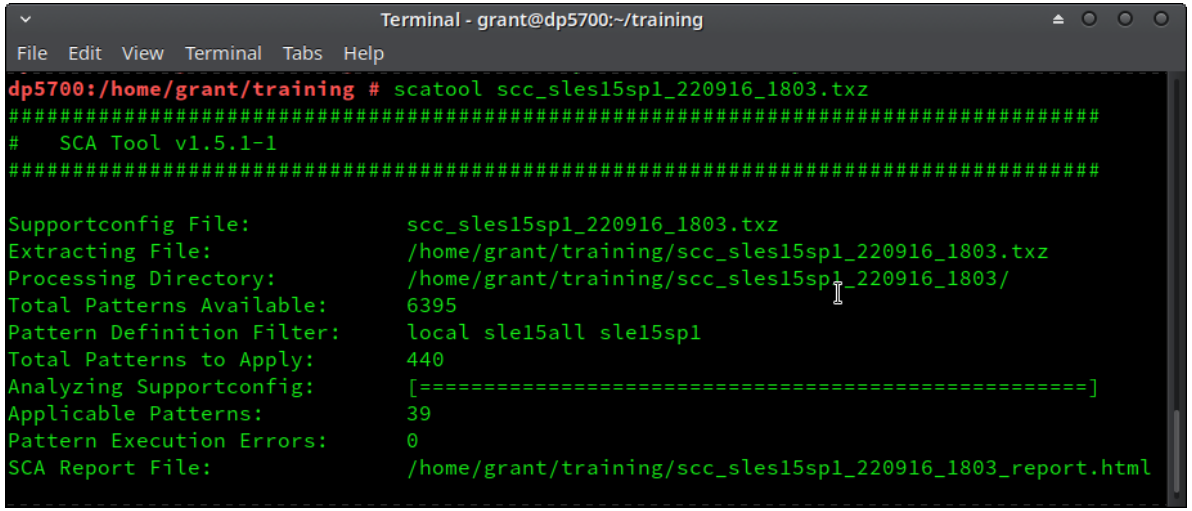
- `getappcore` can be used against a core dumped to the PID's working directory (traditional dump location) or stored in systemd journal (standard location of application dumps on systemd).

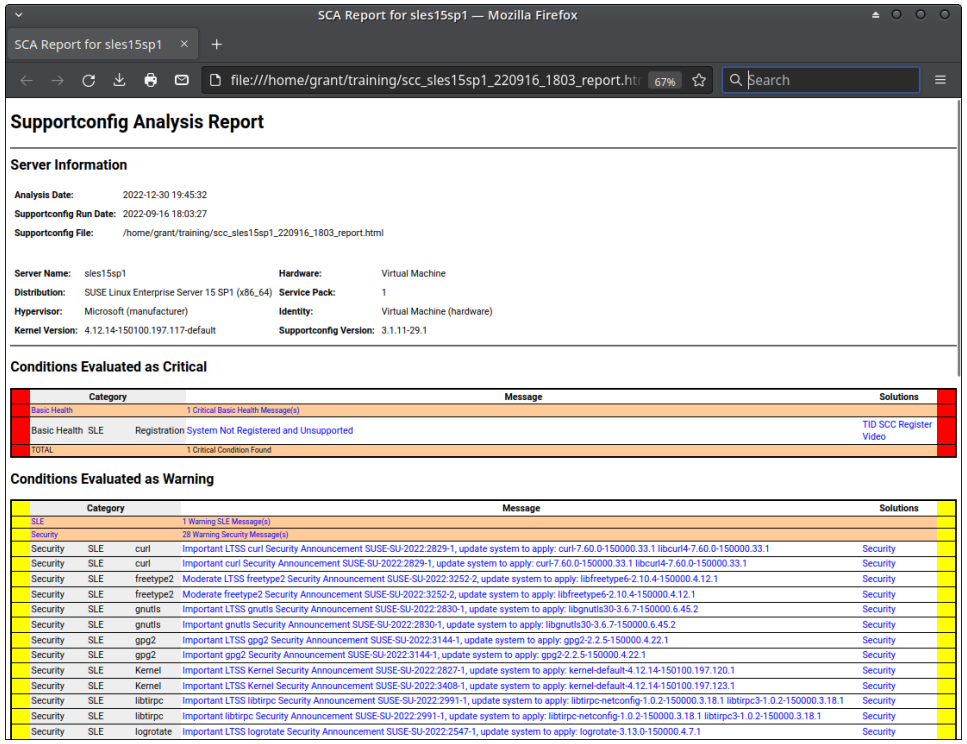
```
linux-tbkh:/var/log # echo $$
2314
linux-tbkh:/var/log # bash
linux-tbkh:/var/log # echo $$
3686
linux-tbkh:/var/log # kill -11 3686
Segmentation fault (core dumped)
linux-tbkh:/var/log # coredumpctl list
TIME                            PID   UID   GID SIG PRESENT EXE
Fri 2023-01-06 15:22:29 PST    2355     0     0  11 * /bin/bash
Fri 2023-01-06 15:26:57 PST    3686     0     0  11 * /bin/bash
linux-tbkh:/var/log # getappcore -j 3686
################################################################################
Get Application Core Tool, v1.52.8_dev2
Date:     01/06/23, 15:27:15
Server:   linux-tbkh
OS:       SUSE Linux Enterprise Server 12 SP3
Kernel:   4.4.180-94.164-default (x86_64)
Corefile: PID 3686
################################################################################

Local configuration file... None
Retrieving core file with PID 3686... Done
Validating core file... Done
Validating binary file... Done
Checking Source Binary with chkbin... Done
Building list of required libraries... Done
Building list of required RPMs... Done
Building list of debuginfo RPMs... Warning
 + libreadline6: RPM and SRC version did not match. Debuginfo name may not be reliable! --
 Done
Setting gdb environment variables... Done
Creating gdb startup files... Done
Creating core archive... Done

Affected Binary: /bin/bash
Coredump File:   /var/log/core.3686
Archive Name:    /var/log/scc_linux-tbkh_bash_230106_152715_appcore.txz
Removing required files and directories ... Done

Finished!
linux-tbkh:/var/log # _
```





SCA Tool

- **S**upport**C**onfig **A**nalysis Tool

- Use supportconfig archive as an input file to generate an HTML report. All relevant TIDs and pertinent info linked in resulting report.

- Linux power-users can run it as a stand-alone command

- It can also be configured as an FTP/HTTP server appliance so novice Linux users, Windows, and Mac OS X users can use it too! (as long as someone is willing to manage the appliance as a pet project.)

- Available in the "`sca-server-report`" package. Relevant patterns must also be installed for it to be useful. e.g. `sca-pattern-sle15`

Patterns updated whenever a new TID is published in SUSE's knowledge base.

Questions?

References

Magnifying lens icon (Page 8)
https://publicdomainvectors.org/en/free-clipart/Magnifying-lens-icon/40917.html

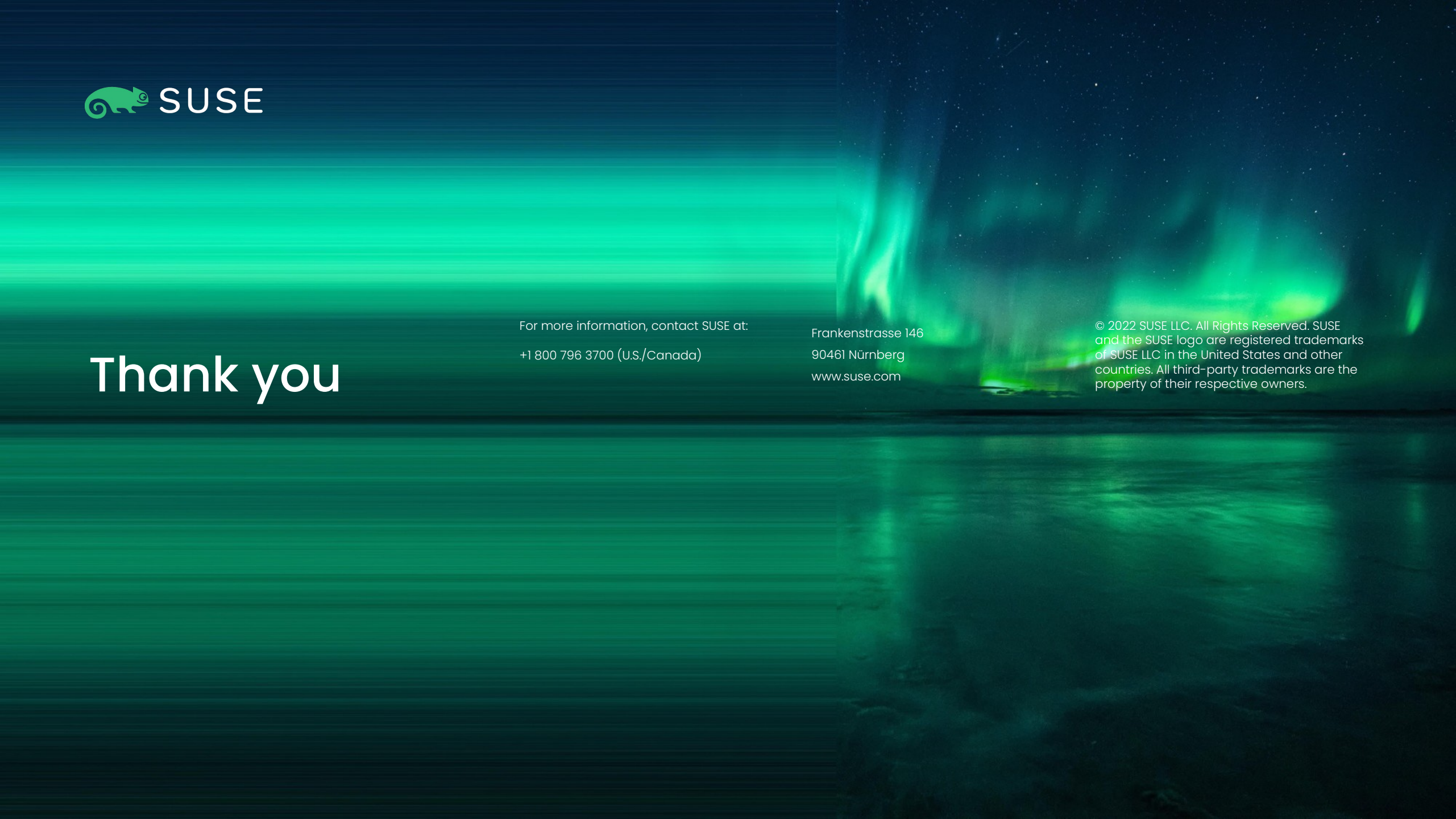
Cloudy Weather Icon (Page 9)
https://publicdomainvectors.org/en/free-clipart/Cloudy-weather-icon/1024.html

Sad computer monitor vector drawing (Page 11)
https://publicdomainvectors.org/en/free-clipart/Sad-computer-monitor-vector-drawing/21950.html

All other images were screencaptures of a KVM DomU in virt-manager

About Debugging Symbols (*-debug-info package contents):
https://tldp.org/LDP/lfs/5.0/html/chapter06/aboutdebug.html

Recommended Listening
Logical Song – Supertramp – Breakfast in America – 1979



Thank you

For more information, contact SUSE at:

+1 800 796 3700 (U.S./Canada)

Frankenstrasse 146

90461 Nürnberg

www.suse.com