Richtlinie zum „Software-Sprint“

ncp2 – Nextcloud Atomic

Schlussbericht

Zuwendungsempfänger:

Tobias Knöppler

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS24S26 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Nextcloud füllt als Plattform für digitale Kollaboration eine wichtige Lücke in der Open-Source-Landschaft. Der Betrieb von Nextcloud erfordert jedoch einiges an Aufwand und Fachwissen, insbesondere wenn er modernen Sicherheitsanforderungen genügen muss.

Deshalb wollte ich mit Nextcloud Atomic ein vollständiges System entwickeln, dass alle zentralen administrativen Aufgaben für den Betrieb einer Nextcloud-Instanz auf unkomplizierte Weise abbildet und das Hosting damit einer größeren Zielgruppe zugänglich macht, ohne dass diese dabei Kompromisse bei Sicherheit und Stabilität in Kauf nehmen muss.

Kern des Ansatzes sollte eine Linuxdistribution mit unveränderlicher System-Partition (sog. Immutable distribution) bieten, die Systemupdates in Form von Partitionsabbildern erhält, welche die vorhergehende Version beim Update vollständig ersetzen.
Dieser Ansatz für Updates sorgt dafür, dass sich die Systeme (abgesehen von Konfiguration und User-Daten) bytegenau gleichen, egal ob frisch installiert oder seit Jahren geupdatet und macht es deutlich leichter, die Funktionalität über Versionen hinweg sicherzustellen.

Auf dieser Linuxdistribution sollte sodann das offizielle Multi-Container-Deployment von Nextcloud, Nextcloud All-in-One installiert werden, sodass Nextcloud Atomic die Konfiguration von Nextcloud selbst dem Upstream-Projekts überlassen und sich auf die Integration der Services und die Administration des Betriebsystems fokussieren kann.

Diese sollte in Form einer eigens entwickelten Weboberfläche von Administratoren konfiguriert und verwaltet werden können, welche unter anderem Funktionen zum Management von Backups, Festplatten, automatischen System- und Nextcloud-Updates und DynDNS bereitstellt.

Beitrag des Projektes zu den Zielen der Förderinitative „Software-Sprint“

Nextcloud ist als Open-Source-Plattform für digitale Zusammenarbeit ein wichtiger Baustein für die Unabhängigkeit insbesondere kleiner und gemeinnütziger Organisationen, aber auch Privatpersonen von großen, amerikanischen Tech-Unternehmen und für die Wahrung der Hoheit über die eigenen Daten.

Mit Nextcloud Atomic wird der Betrieb insbesondere für diese Zielgruppen günstiger, sicherer und zugänglicher.

Ausführliche Darstellung der Ergebnisse

Ergebnisse

Das zugrundeliegende Betriebssystem für Nextcloud Atomic mit unveränderlicher System-Partition wurde auf Basis des Build-Systems mkosi und Debian als Grundlage implementiert. Besonderer Wert wurde darauf gelegt, dass die kryptographische Integrität des Systems beim Booten festgestellt und daran die Bereitstellung der Schlüssel für die verschlüsselte Datenpartition, sowie sämtliche Passwörter für Systemdienste geknüpft ist (Trusted Boot mit TPM2 als Secret Provider).

Das Betriebssystem muss nicht erst installiert werden, sondern wird automatisch beim ersten Hochfahren eingerichtet und legt alle erforderlichen Partitionen an, verschlüsselt diese wo erforderlich und erstellt benötigte Dateien.

Nach dem ersten Start ist ein Setup-Wizard als Weboberfläche verfügbar, über welchen die Initialpasswörter und die Domain, unter welcher Nextcloud erreichbar sein soll, konfiguriert werden.

Im Zuge dessen wurden minimale (i.d.r. grpc-) Schnittstellen zwischen den Systemdiensten eingerichtet, sodass z.B. der Setup-Wizard Zugriff auf das Kommandozeilenwerkzeug von Nextcloud erhält, ohne direkten Dateizugriff auf die Dateien von Nextcloud zu benötigen.

Nextcloud All-in-One läuft mit grundlegenden Funktionen inklusive Podman als Container-Runtime unter einem unprivilegierten User innerhalb eines Systemdienstes, der vom Host-System weitgehend abgeschottet ist und lediglich Zugriff auf die minmal erforderlichen Dateien und Schnittstellen hat.

Von außen erreichbar ist weder Nextcloud noch das Admin-/Setup-Webinterface direkt, sondern Caddy als Reverse-Proxy (als weiterer Systemdienst), welcher im Laufe des Setup-Prozesses oder während Wartungsarbeiten dynamisch umkonfiguriert wird, um jeweils die benötigten Services von außen erreichbar zu machen. Caddy erstellt und erneuert außerdem automatisch selbst signierte (für IP-Adressen und lokale Domains) oder von Let’s Encrypt signierte TLS-Zertifikate.

Die Konfiguration und teilweise das Dateisystem für die Systemdienste ist als Images (Systemd System extensions oder Service Root Images) bereitgestellt, die unabhängig von der Systempartition aktualisiert werden können, sodass nur ein Minimum an Diensten bei Updates neugestartet werden muss.

Nicht erreichte Meilensteine

Einige Meilensteine konnten im Projektzeitraum nicht erreicht werden.

So konnte die Arbeit am Admin-Interface die Unterstützung für automatische Updates nicht fertiggestellt werden und die Implementation der Backup-Funktionalität fehlt noch.

Auch wird zum jetztigen Zeitpunkt nur das bauen von VM-Images (incus-VMs) für die amd64-Prozessorarchitektur unterstützt und noch keine konkreten Hardwareplattformen (geplant war Unterstützung für den Raspberry Pi) und es sind noch keine Images zum Download verfügbar.

Erkenntnisse

Ursprünglich hatte ich geplant, nicht nur Nextcloud All-in-One, sondern auch diverse andere Dienste, z.B. Caddy und das Admin-Interface, in Podman-Containern laufen zu lassen. Bei der ausführlichen Beschäftigung mit der Systemd-Service-Konfiguration erkannte ich jedoch, dass Systemd sämtliche Funktionen, für die ich Container einsetzen wollte, bereits selbst unterstützt und mir deutlich bessere Kontrolle darüber und dabei eine bessere Integration in das Service-Management bietet, weshalb am Ende nur die Upstream-Services in OCI-Containern laufen.

Überrascht war ich davon, dass systemd-sysupdate – die Lösung, die bei Nextcloud Atomic für Image-basierte Updates eingesetzt wird – zwar die Konfiguration der Updates selbst erlaubt, jedoch keine Funktionen bietet, um betroffene Services neu zu starten oder zu laden. Dies musste also von mir selbst implementiert werden.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Die eigentliche Zielgruppe wird aus den Ergebnissen des Projekts erst in größerem Umfang profitieren, wenn es fertige Images zum Download gibt. Die hierzu nötigen Entwicklungen sind aber nach Projektabschluss überschaubar.

Grundsätzlich ist Nextcloud Atomic gut geeignet, um als Referenz für die Verwendung von Linux-Distributionen mit unveränderlicher Systempartition als moderne, modulare Server-Systeme zu dienen. Die hierbei verwendeten Technologien sind sehr neu und es gibt bislang nur wenige Beispiele für die Nutzung.

Ich werde das Projekt auf Basis des aktuellen Ergebnisses bis zur Fertigstellung der ursprünglich geplanten Meilensteine weiterentwickeln und anschließend um weitere Funktionen ergänzen (Infrastruktur zum automatischen Bauen, Testen und Release von Images, Unterstützung für Metriken für Monitoring & Alerting, Unterstützung von Systemen ohne TPM2-Chip, Images für Raspberry Pi, amd64, arm64, qemu und diverse cloud provider, uvm.).

Ich selbst konnte bei der Arbeit an Nextcloud Atomic sehr umfangreiches Wissen über die Funktionsweise von Linux-Betriebssystemen und deren Komponenten (insbesondere den Systemd-Stack) aufbauen, wovon zukünftige Projekten mit Sicherheit profitieren werden.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Ursprünglich hatte ich nach Evaluierung diverser Optionen die Entwicklung des Betriebsystems mit Buildroot und SkiffOS als Build-System begonnen, musste aber im Laufe der Entwicklung feststellen, dass durch den Fokus dieses Build-Systems auf Embedded Devices einige Funktionen die für meinen Anwendungsfall zentral sind, keine ausreichende Unterstützung erhielten - so zum Beispiel Trusted Boot, Festplattenverschlüsselung, zeitnahe Security-Patches und Container-Runtimes. Dies hätte die

Verantwortung für und die Aufwände durch die Wartung der entsprechenden Komponenten von Buildroot auf Nextcloud Atomic verlagert und damit viele Entwicklungskapazitäten gebunden. Deshalb habe ich entschieden, von SkiffOS auf mkosi zu migrieren, obwohl ich bereits einige Monate auf der ursprünglichen Basis gearbeitet hatte. Dabei konnte ich viele Erkenntnisse, aber wenig Code wiederverwenden.

Anfänglich hatte ich Nextcloud All-in-One mit Docker als Container-Engine konfiguriert, stieß dabei aber immer wieder an Grenzen bei der Integration in das Service-Management des Betriebssystems und bei der Isolation der Containerprozesse. Ursache für die Probleme war die Architektur von Docker, bei der alle Container-Prozesse von einem zentralen Daemon-Prozess gestartet werden. Das hat zur Folge, dass alle Container-Prozesse die Prozessgruppe des Docker-Daemons erben anstatt die des Service-Prozesses, der das Starten des Containers auslöst. In der Folge lassen sich viele der Sicherheitsfunktionen von Systemd nicht gut auf Docker-Container anwenden. Zudem ist es nicht möglich, Container unter mehreren (unprivilegierten) Usern auszuführen, da es nur einen Docker-Daemon-Prozess geben kann. Diese Gründe veranlassten mich schließlich, von Docker zu Podman zu migieren, obwohl Nextcloud All-in-One offiziell nur Docker unterstützt. Glücklicherweise ist Podman größtenteils kompatibel mit der docker- und docker-compose-Spezifikation, sodass ich hierbei nicht auf größerer Probleme stieß - außer ein paar Anpassungen, die ich an der docker-compose.yml vornehmen musste. Hierfür habe ich ein Skript geschrieben, welches als Teil des Build-Prozesses die docker-compose.yml herunterlädt, korrigiert und schließlich in das System-Image ablegt.

Bei der Implementation des Partitionslayouts stieß ich auf einen Bug in systemd-repart, welcher es nicht möglich machte, das von mir ursprünglich geplante Partitionslayout (readonly /usr-Partition, ephemeral (d.h. beim Reboot zurückgesetzte) root-Partition, verschlüsselte /var-Partition) umzusetzen. Stattdessen habe ich das Partitionslayout nun mit einer verschlüsselten root-Partition und ohne extra /var-Partition implementiert und werde dieses Layout anpassen, sobald der besagte Bug behoben ist.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Die zentrale Anlaufstelle für Informationen zu Nextcloud Atomic ist die Projekt-Webseite und -Dokumentation: https://nextcloudatomic.com

Die zugehörigen Repositories, welche den Quellcode des Projekts enthalten sind unter https://github.com/nextcloud-atomic zu finden.

Die beste Möglichkeit, um über Updates, Releases und Artikel zu Nextcloud Atomic benachrichtigt zu werden, ist der offizielle Mastodon-Account des Projekts: @nextcloudatomic@mastodon.social

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Im Projektverlauf gab es keine Ereignisse, die eine Anpassung der Kostenplanung erforderlich machten.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Der Entwickler von Nextcloud All-in-One schlug vor, die Backup-Funktionalität von Nextcloud Atomic mit der integrierten von Nextcloud All-in-One kompatibel zu machen, was unter anderem bedeuten

würde statt der angedachten Lösung, Kopia, die von Nextcloud AiO verwendetete, Borg, zu verwenden. Dies hatte jedoch keinen Einfluss auf das Projekt, da die Arbeit an der Backupfunktionalität noch nicht begonnen wurde.