The unbelievable Machine Company

NERD STUFF #4

Session Manager: Zugriff auf On-prem-Hardware via AWS

Ein Use Case von Christian Hufgard,

The unbelievable Machine Company

Klingt verrückt? Zugegeben, AWS zu nutzen um On-Prem-Hardware zu managen ist sicherlich ein spezieller Anwendungsfall, aber es gibt durchaus Situationen, in denen das sinnvoll sein kann. Welche Vor- und Nachteile es gibt und wie es geht, werden im Folgenden ausgeführt:

1. Vorteile

AWS bietet zahlreiche Werkzeuge an, um Server zu verwalten und sie vor allem in einem definierten Zustand zu halten. Für Maschinen, die direkt bei AWS gehostet sind (EC2), funktioniert das natürlich am besten. Sie sind nahtlos integriert. Aber die dafür nutzbaren Werkzeuge wurden in den letzten Jahren immer mehr erweitert und ermöglichen jetzt – und das ist der größte Vorteil – das zentrale Verwalten des gesamten Maschinen-Pools. Das Definieren und Ausrollen von Patch-Baselines über den Systems Manager gehört genauso dazu wie das zentrale Sammeln von Logfiles über CloudWatch.

1.1 Patch Baseline

Mit einer Patch-Baseline kann ein Administrator definieren, welche Patches für die Installation auf den verwalteten Instanzen genehmigt sind. Das funktioniert sowohl durch manuelle als auch durch automatische Freigabe für z.B. wichtige Updates. Zusätzlich gibt es vordefinierte Patch-Baselines für jedes unterstützte Betriebssystem.

1.2 Session Manager

Ein ebenfalls sehr interessantes Feature des Systems Manager ist der Session Manager. Über den kann sich ein User direkt auf einem Server eine Session öffnen, ohne dass hierfür ein SSH-Server laufen muss. Hierdurch entfällt die Notwendigkeit, SSH-Schlüssel auf den Maschinen zu verteilen und ggf. wieder zu entfernen. Der Zugriff kann direkt über IAM zentral kontrolliert werden.

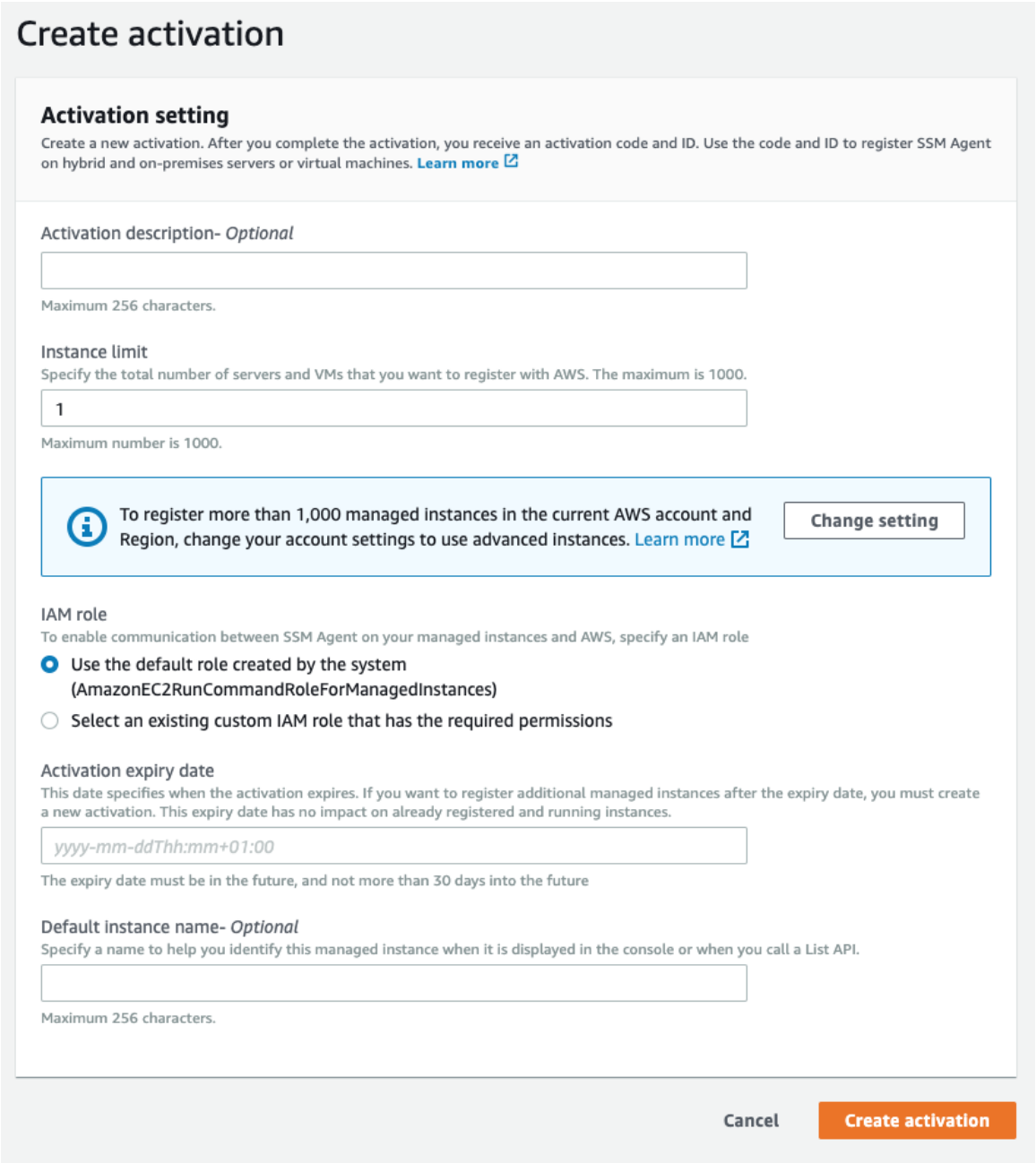
2. Wie geht's?

2.1 Session Manager konfigurieren

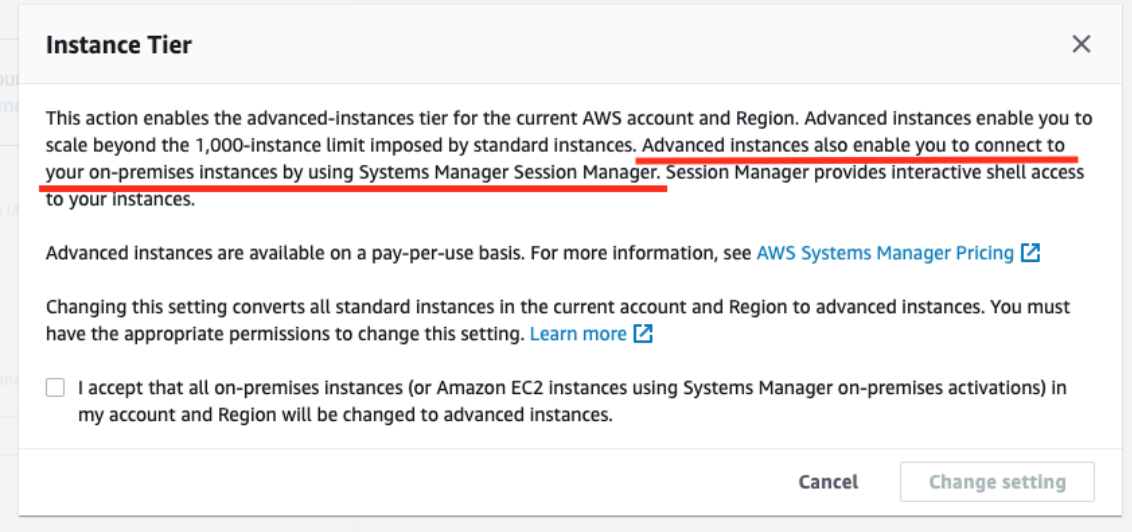
Eine ausführliche Anleitung zum Aufbau einer hybdriden Cloud-Lösung befindet sich unter https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-

managedinstances.html. Da es in diesem Beitrag nur um den Session Manager geht, werden einige andere wichtige (!) Schritte ausgelassen.
Die ganze Installation kann wahlweise über die AWS CLI oder die AWS Console durchgeführt werden. Der Einfachheit halber wird hier nur der Weg über die Console vorgestellt.

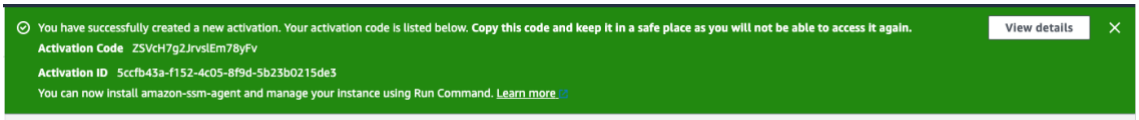
Nach dem Öffnen des Systems Manager wählt man, wenig überraschend, die Option „Hybrid Activations“ aus.



Der einzige wirklich wichtige Schritt ist – für einen Test –, auf „advanced instances“ umzustellen.



Hat man dann auf „Create Activation“ geklickt, erhält man Activation-Code und ID:



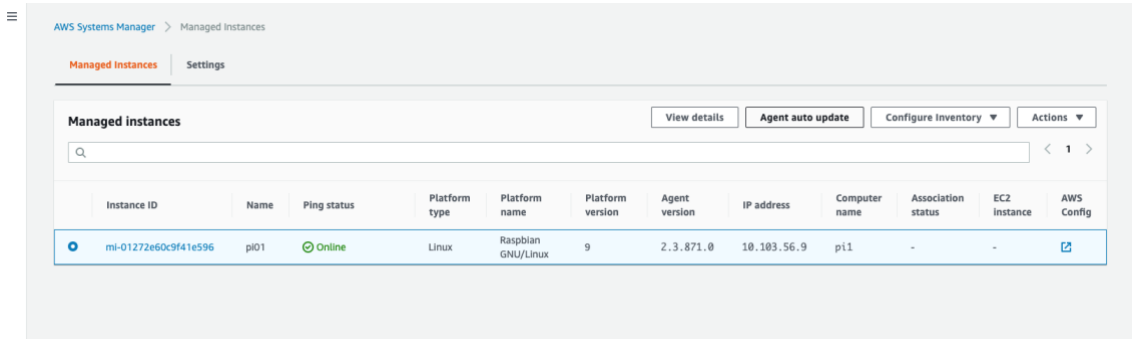
Der erste Schritt ist getan, weiter geht es auf dem Device selbst. Ich arbeite mit einem Raspberry Pi, der direkt vor mir auf dem Schreibtisch steht.

2.2 On-prem Hardware konfigurieren

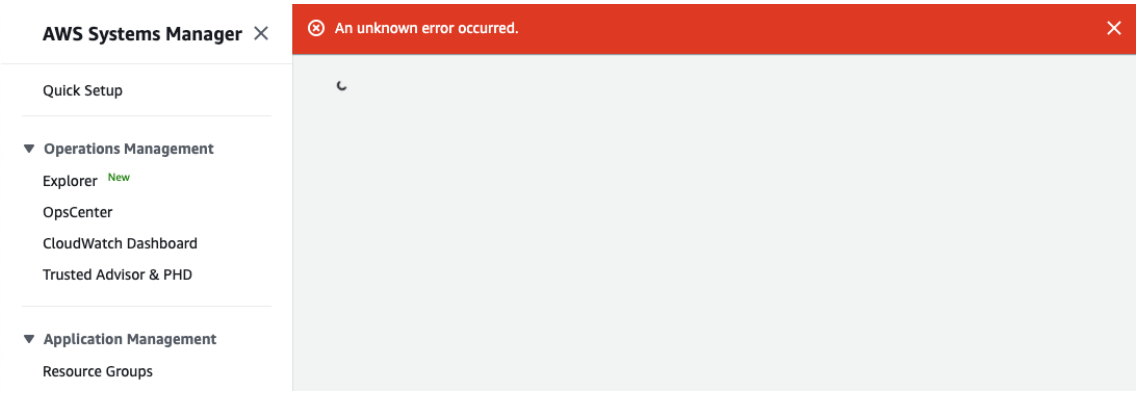
Auf dem Pi werden folgende Befehle ausgeführt:

```
mkdir /tmp/ssm
sudo curl https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/debian_arm/amazon-ssm-agent.deb -o /tmp/ssm/amazon-ssm-agent.deb
sudo dpkg -i /tmp/ssm/amazon-ssm-agent.deb
sudo service amazon-ssm-agent stop
sudo amazon-ssm-agent -register -code "activation-code" -id "activation-id" -region "region"
sudo service amazon-ssm-agent start
```

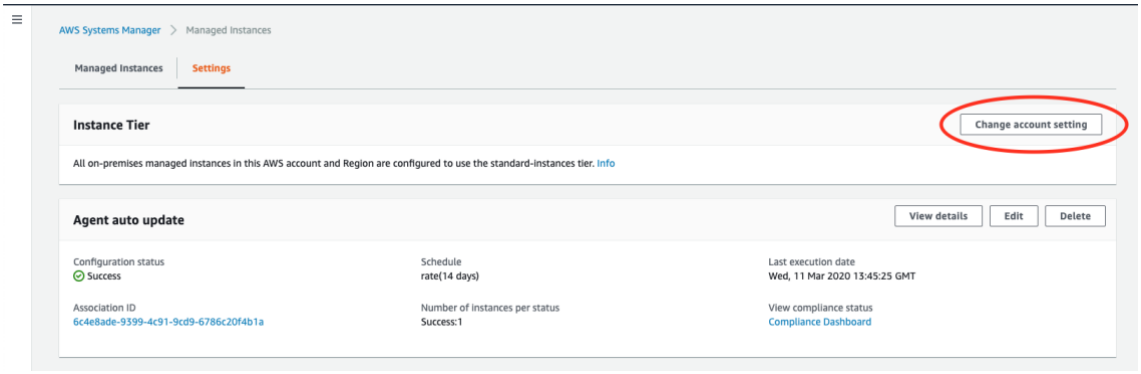
Nach einigen Minuten taucht er dann bei den Managed Instances auf:



Jetzt den Pi auswählen und unter „Actions“ „Start Session“ auswählen. Hat man vergessen „Advanced Instances“ auszuwählen, gibt es eine absolut nichtssagende Fehlermeldung:



Dies kann aber noch nachträglich unter „Settings“ geändert werden:



Spätestens jetzt sollte das Starten der Session funktionieren:

2.3 Zugriff über SSH

Man möchte sich nicht unbedingt erst bei der AWS-Console anmelden, um von da aus eine Session auf einem Gerät aufzumachen. Deshalb ist es auch möglich, direkt auf einem Terminal eine SSH-Verbindung zu tunneln. Die Schritt-für-Schritt-Anleitung befindet sich hier https://docs.aws.amazon.com/systems-manager/latest/userguide/session-manager-working-with-install-plugin.html

Hierfür muss man vorher das Session Manager Plugin für die AWS CLI installieren. Vergisst man diesen Schritt, gibt es immerhin einen entsprechenden Hinweis:

```
SessionManagerPlugin is not found. Please refer to SessionManager
Documentation here: http://docs.aws.amazon.com/console/systems-
manager/session-manager-plugin-not-found
```

Danach wird .ssh/config angepasst:

```
# SSH over Session Manager
host i-* mi-*
ProxyCommand sh -c "aws ssm start-session --target %h --document-name AWS-
StartSSHSession --parameters 'portNumber=%p'"
```

Und schon ist ein Login per ssh über die Instance-ID möglich. Hierfür müssen aller-dings wieder persönliche Credentials verwendet werden, da der Session Manager Agent sich mit dem SSH-Daemon verbindet:

```
MAC-UM-020:~ christian.hufgard$ ssh mi-01272e60c9f41e596
christian.hufgard@mi-01272e60c9f41e596's password:
```

Auch muss der SSH-Daemon laufen. Allerdings kann er so umkonfiguriert werden, dass nur Zugriff von localhost erlaubt ist, da sich der Systems Manager Agent als Proxy direkt über localhost verbindet.

Wie sich an der SSH-Konfiguration erkennen lässt, ist auch ein Zugriff auf EC2-Instanzen möglich. Man kann sich also den Bastion-Host sparen, wenn man den Agent nachinstalliert oder ein AMI verwendet, bei dem er vorinstalliert ist. Konkret sind dies:

- Windows Server 2008-2012 R2 AMIs published in November 2016 or later
- Windows Server 2016 and 2019
- Amazon Linux
- Amazon Linux 2
- Ubuntu Server 16.04
- Ubuntu Server 18.04
- Amazon ECS-Optimized

3. Nachteile

Als erster und offensichtlichster Nachteil sind die Kosten zu nennen. Advanced Instances kosten aktuell 0,00695 Dollar pro Minute Laufzeit. Bei einer 24x7 laufenden Maschine sind das 5 Dollar pro Monat. Die bekannte Software SSH-Daemon wird gegen den eher unbekannten Systems Manager Agent ausgetauscht. Der wird allerdings, sofern man nichts an der Konfiguration ändert, alle 14 Tage automatisch aktualisiert, so dass man sich um Sicherheits-Updates etwas weniger Sorgen machen muss als beim SSH-Daemon. Dieser lässt sich wiederum auch zum Beispiel über einen Cron-Job aktualisieren.

AWS rät übrigens dazu, den Systems Manager Agent häufiger als alle 14 Tage automatisch aktualisieren zu lassen! Fällt die Hardware aus oder wird sie abgeschaltet, dauert dies auch wieder einige Minuten, bis der Systems Manager den ausbleibenden Ping des Agents meldet. Wobei in einer Produktivumgebung (hoffentlich) ohnehin zusätzliches Monitoring konfiguriert ist.

4. Fazit

On-Prem-Hardware lässt sich, ein unterstütztes OS vorausgesetzt, einfach und zentral über AWS steuern. Will man weitere Sicherheitsfunktionen nutzen – zum Beispiel das Abschalten des SSH-Zugriffs von außen –, muss man dafür bezahlen, was ansonsten erst ab mehr als 1.000 verwalteten Instanzen notwendig ist.

Autor

Christian Hufgard ist Data Architect in Applications am *um Standort Frankfurt. In seiner Freizeit ist er 1. Vorsitzender eines Freifunk-Vereins. Sein bisher größtes Freifunk-Projekt bestand aus einer Installation auf dem „Hessentag“, dem größten und ältesten Landes-fest Deutschlands. In der Spitze waren über 900 Nutzer in das Netzwerk eingeloggt.

Kontakt

The unbelievable Machine Company GmbH
Michelle Zirnsak (Marketing & Sales)
Grolmanstr. 40
D-10623 Berlin
+49 (0) 30 88926560

whitepaper@unbelievable-machine.com
www.unbelievable-machine.com