The unbelievable Machine Company

NERD STUFF #4

Session Manager: Accessing On-Premises Hardware via AWS

A Use Case by Christian Hufgard,
The unbelievable Machine Company

Sounds crazy? Admittedly, using AWS to manage on-prem hardware is certainly a special use case, but there are situations where it can be useful. The advantages and disadvantages and how to make it work are being discussed below:

1. Advantages

AWS offers numerous tools to manage servers and especially to keep them in a defined state. Of course this works best for machines hosted directly at AWS (EC2). They are integrated seamlessly. But the tools that can be used for this purpose have been extended more and more over the last few years and now allow – and this is the biggest advantage – the central management of the entire pool of machines. Defining and rolling out patch baselines via the Systems Manager is just as much a part of this as the central collection of log files via CloudWatch.

1.1 Patch Baseline

A patch baseline allows the administrator to define which patches are approved for installation on managed instances. This can be done both by manual and automatic approval for, e.g. important updates. Additionally, there are predefined patch baselines for each supported operating system.

1.2 Session Manager

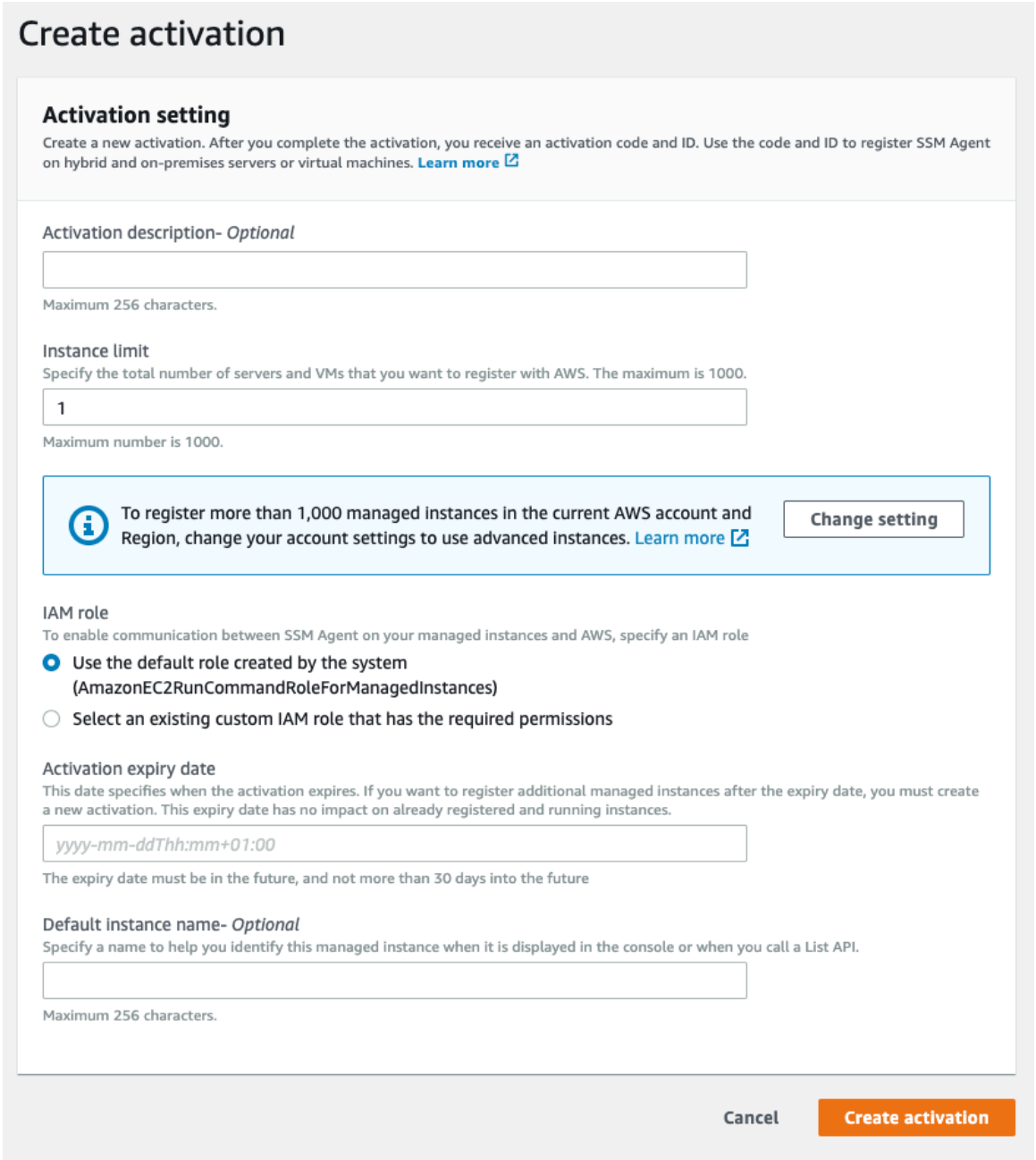
Another very interesting feature of the Systems Manager is the Session Manager. It allows the user to open a session directly on a server without having to run an SSH server. This eliminates the need to distribute SSH keys onto the machines and remove them again if necessary. Access can be controlled centrally directly via IAM.

2. How does it work?

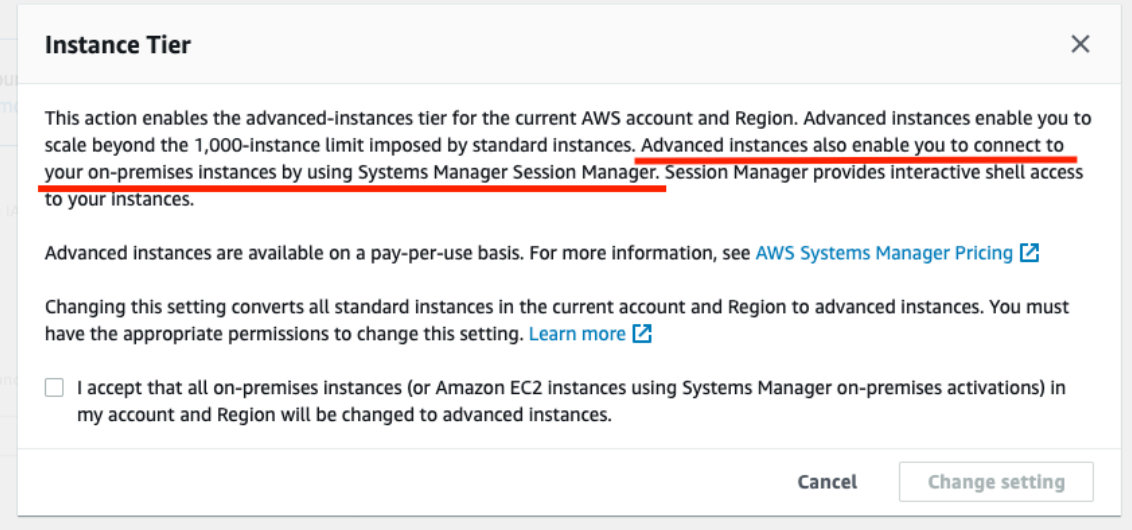
2.1 Configurating the Session Manager

A detailed guide to building a hybrid cloud solution can be found at https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-managedistances.html. Since this article is only about the Session Manager, some other important (!) steps will be left out.
The whole installation can be done either via the AWS CLI or the AWS Console. To keep things simple, only the way via the console is presented here.

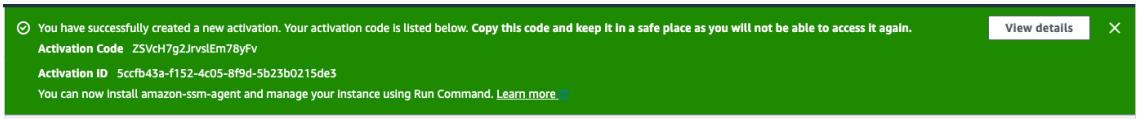
After opening the System Manager, you select, not surprisingly, the option "Hybrid Activations".



The only really important step - for a test - is to switch to "advanced instances".



Once you have clicked on "Create Activation", you will receive the activation code and ID:



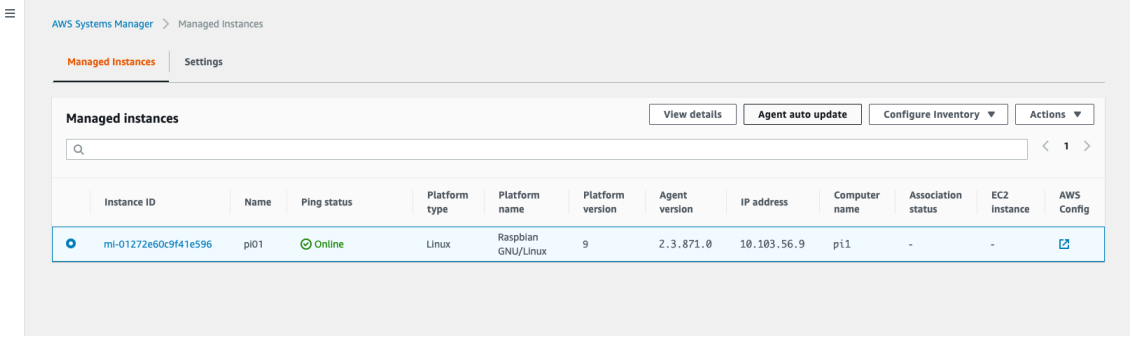
Now that the first step is done, you can continue on the device itself. I work with a Raspberry Pi, which is standing right in front of me on the desk.

2.2 Configurating On-prem Hardware

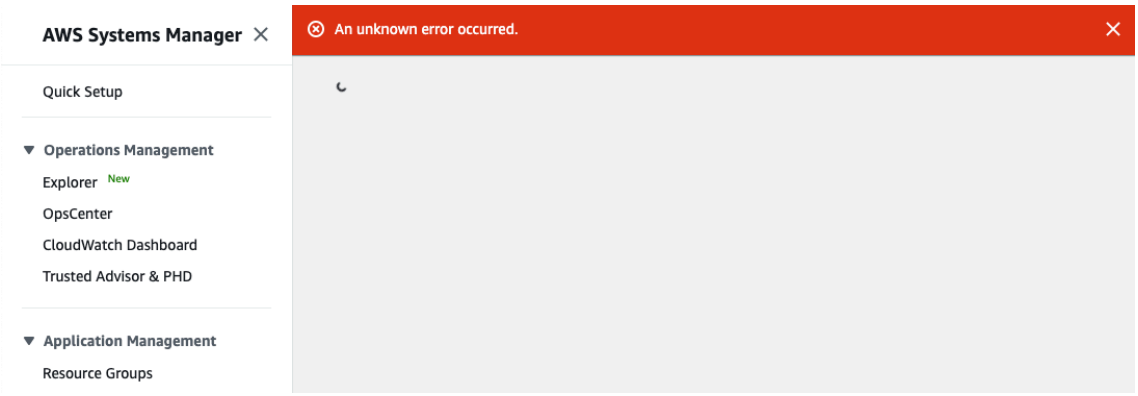
The following commands are executed on the Pi:

```
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
```

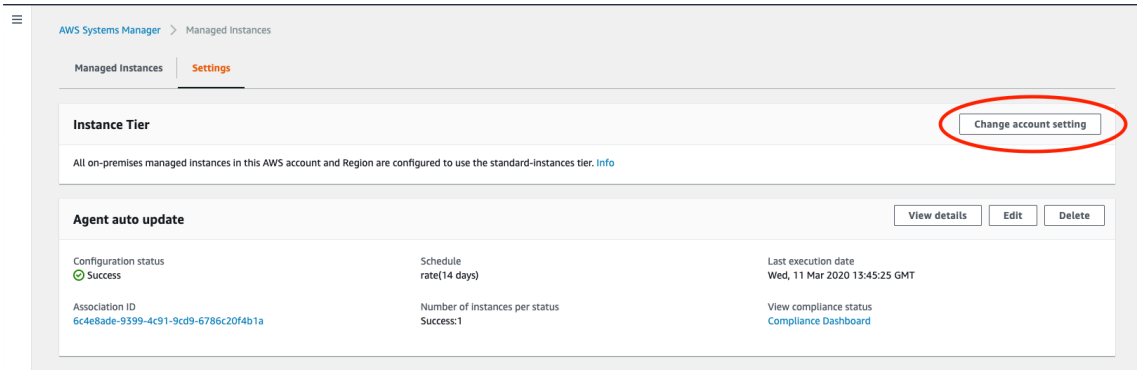
Within a couple of minutes, it will appear in the Managed Instances:



Now select the Pi and choose "Start Session" under "Actions". If you forgot to select "Advanced Instances", there will be a completely meaningless error message:



However, this can be corrected under "Settings":



At this point, starting the session should work:

2.3 Access via SSH

You don't necessarily want to log in to the AWS console first to open a session on a device from there. Therefore it is also possible to tunnel an SSH connection directly on a terminal. The step-by-step instructions can be found here https://docs.aws.amazon.com/systems-manager/latest/userguide/session-manager-working-with-install-plugin.html.

To do this you must first install the Session Manager plugin for the AWS CLI. If you forget this step, there will be a corresponding message:

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

Afterwards .ssh/config will be adjusted:

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

Now you can login via ssh using the instance ID. However, personal credentials must be used for this, since the Session Manager Agent connects to the SSH daemon:

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

Additionally, the SSH daemon must be running. However, it can be reconfigured to allow access from localhost only, since the Systems Manager Agent connects directly via localhost as a proxy.

As can be seen from the SSH configuration, access to EC2 instances is also possible. So you can save yourself the bastion host if you install the agent later or use an AMI where it is pre-installed. These are specifically:

- 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. Disadvantages

The first and most obvious disadvantage is the cost. Advanced Instances currently cost $ 0.00695 per minute of runtime. For a 24 x 7 running machine, this is $ 5 per month. The well-known software SSH daemon is replaced by the rather unknown Systems Manager Agent. However, unless you change the configuration, it will be updated automatically every 14 days, so you don't have to worry as much about security updates as with the SSH daemon. This daemon can also be updated via cron job, for example.

By the way, AWS advises to let the Systems Manager Agent update automatically more often than every 14 days! If the hardware fails or is switched off, it will take several minutes until the Systems Manager reports the missing ping of the agent. Whereby in a production environment (hopefully) additional monitoring is configured anyway.

4. Conclusion

On-Prem hardware can be easily and centrally controlled via AWS, given a supported OS. If you want to use additional security functions – for example, switching off SSH access from the outside - you have to pay for what is otherwise only necessary with more than 1,000 managed instances.

Author

Christian Hufgard is Data Architect in applications at the *um location Frankfurt. In his spare time he is 1st chairman of a free radio association. His biggest free radio project so far was an installation at the "Hessentag", the biggest and oldest state festival in Germany. At the peak more than 900 users were logged into the network.

Contact

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