Article
Mark Bolinsky · Mar 21, 2017   4m read

InterSystems IRIS and Caché Application Consistent Backups with Azure Backup

Database systems have very specific backup requirements that in enterprise deployments require forethought and planning. For database systems, the operational goal of a backup solution is to create a copy of the data in a state that is equivalent to when application is shut down gracefully. Application consistent backups meet these requirements and Caché provides a set of APIs that facilitate the integration with external solutions to achieve this level of backup consistency.

These APIs are *ExternalFreeze* and *ExternalThaw*. *ExternalFreeze* temporarily pauses writes to disk and during this period Caché commits the changes in memory. During this period the backup operation must complete and be followed by a call to *ExternalThaw*. This call engages the write daemons to write the cached updated in the global buffer pool (database cache) to disk and resumes normal Caché database write daemon operations. This process is transparent to user processes with Caché. The specific API class methods are:

##Class(Backup.General).ExternalFreeze()

##Class(Backup.General).ExternalThaw()

These APIs in conjunction with the new capability of Azure Backup to execute a script prior and after the execution of a snapshot operation, provide a comprehensive backup solution for deployments of Caché on Azure. The pre/post scripting capability of Azure Backup is currently available only on Linux VMs.

Prerequisites

At the high level, there are three steps that you need to perform before you can backup a VM using Azure Backup:

1. Create a Recovery Services vault
2. Install has the latest version of the VM Agent.
3. Check network access to the Azure services from your VM.

The Recovery Services vault manages the backup goals, policies and the items to protect. You can create a Recovery Services vault via the Azure Portal or via scripting using PowerShell. Azure Backup requires an extension that runs in your VM, is controlled by the Linux VM agent and the latest version of the agent is also required. The extension interacts with the external facing HTTPS endpoints of Azure Storage and the Recovery Services vault. Secure access to those services from the VM can be configured using a proxy and network rules in an Azure Network Security Group.

For more information about these steps visit Prepare your environment to back up Resource Manager-deployed virtual machines.

Pre and Post Scripting Configuration

The ability to call a script prior to the backup operation and after is, included in the latest version of the Azure Backup Extension (Microsoft.Azure.RecoveryServices.VMSnapshotLinux). For information about how to install the extension please check the detailed feature documentation.

By default, the extension included sample pre and pot scripts located in your Linux VM at

```
/var/lib/waagent/Microsoft.Azure.RecoveryServices.VMSnapshotLinux-1.0.9110.0/main/tempPlugin
```

And needs to be copied to the following locations respectively.

```
/etc/azure/prescript.sh
/etc/azure/postScript.sh
```

You can also download the script template from GitHub.

For Caché, the prescript.sh script where a call to the ExternalFreeze API can be implemented and the postScript.sh should contain the implementation that executes ExternalThaw.

The following is a sample *prescript.sh* implementation for Caché.

```bash
#!/bin/bash
# variables used for returning the status of the script
success=0
error=1
warning=2
status=$success
log_path="/etc/preScript.log"   #path of log file
printf  "Logs:\n" > $log_path
# TODO: Replace <CACHE INSTANCE> with the name of the running instance
csession <CACHE INSTANCE> -U%SYS
"##Class(Backup.General).ExternalFreeze()" >> $log_path
status=$?
if [ $status -eq 5 ]; then
echo "SYSTEM IS FROZEN"
printf  "SYSTEM IS FROZEN\n" >> $log_path
elif [ $status -eq 3 ]; then
echo "SYSTEM FREEZE FAILED"
printf  "SYSTEM FREEZE FAILED\n" >> $log_path
status=$error
csession <CACHE INSTANCE> -U%SYS "##Class(Backup.General).ExternalThaw()"
fi

exit $status
```

The following is a sample *postScript.sh* implementation for Caché.

```bash
#!/bin/bash
# variables used for returning the status of the script
success=0
error=1
warning=2
status=$success
log_path="/etc/postScript.log"   #path of log file
printf  "Logs:\n" > $log_path
# TODO: Replace <CACHE INSTANCE> with the name of the running instance
csession <CACHE INSTANCE> -U%SYS "##class(Backup.General).ExternalThaw()"
status=$?
if [ $status req 5]; then
```

```
echo "SYSTEM IS UNFROZEN"
printf  "SYSTEM IS UNFROZEN\n" >> $log_path
elif [ $status -eq 3 ]; then
echo "SYSTEM UNFREEZE FAILED"
printf  "SYSTEM UNFREEZE FAILED\n" >> $log_path
status=$error
fi
exit $status
```

Executing a Backup

In the Azure Portal, you can trigger the first backup by navigating to the Recovery Service. Please consider that the VM snapshot time should be few seconds irrespective of first backup or subsequent backup. Data transfer of first backup will take longer but data transfer will start after executing post-script to thaw database and should not have any impact on the time between pre & post script.

It is highly recommended to regularly restore your backup in a non-production setting and perform database integrity checks to ensure your data protection operations are effective.

For more information about how to trigger the backup and other topics such as backup scheduling, please check Back up Azure virtual machines to a Recovery Services vault.

#Azure #Backup #Best Practices #Cloud #InterSystems Business Solutions and Architectures #Caché #InterSystems IRIS #InterSystems IRIS for Health

Source
URL:https://community.intersystems.com/post/intersystems-iris-and-cach%C3%A9-application-consistent-backups-azure-backup