

# Backup-VCSA-Postgresql-Database

Run python backup script on vCenter Appliance for the postgresql database and copy it to the defined location

## 1. Versions

Version	Date	User	Comment
---------	------	------	---------

Current version:0.0.7

## 2. Inputs

Name	Type	Description
blnNewHost	boolean	Create new host?
sshExistingHost	SSH:Host	Vcenter Server Appliance SSH Host that is registered
sshExistingKeyPairPath	path	Vcenter Server Appliance SSH key file path
sshExistingKeyPairPassword	SecureString	Vcenter Server Appliance SSH key file password
sshNewHost*	string	Vcenter Server Appliance SSH Host FQDN / IP
sshNewPort*	number	Vcenter Server Appliance SSH Port
sshNewUsername*	string	Vcenter Server Appliance SSH Username
sshNewPassword*	SecureString	Vcenter Server Appliance SSH username password
sshNewRootPath	Array/string	Vcenter Server Appliance SSH Root Path
sshNewKeyPairPath*	path	Vcenter Server Appliance SSH key file path
sshNewKeyPairPassword	SecureString	Vcenter Server Appliance SSH key file password

Name	Type	Description
vcsaBackupLocation	string	Vcenter Server Appliance backup full path
vcsaPythonBackupScript	string	Vcenter Server Appliance python backup full path
remoteBackupPath	path	Remote backup server location
remoteHost	string	Remote Host FQDN/IP
remoteUser	string	Remote username
remotePassword	SecureString	Remote user password

\*Parameter is mandatory

### 3. Attributes

Name	Type	Description
error	string	Error output
exitCode	string	Returns 0 if successful, and a negative value if there is a failure
vcsaBackupFullpath	string	The Backup Path
strUniqueDate	string	Unique Date String
keyPairPath	path	KeyPairPath
keyPairPassword	string	KeyPair Password
sshHost	SSH:Host	SSH Host
newSSHremotePath	string	The Remote Path
newSSHpasswordAuthentication	boolean	Sets authentication to password or key file
newSSHcertificatePath	string	Path to the private key

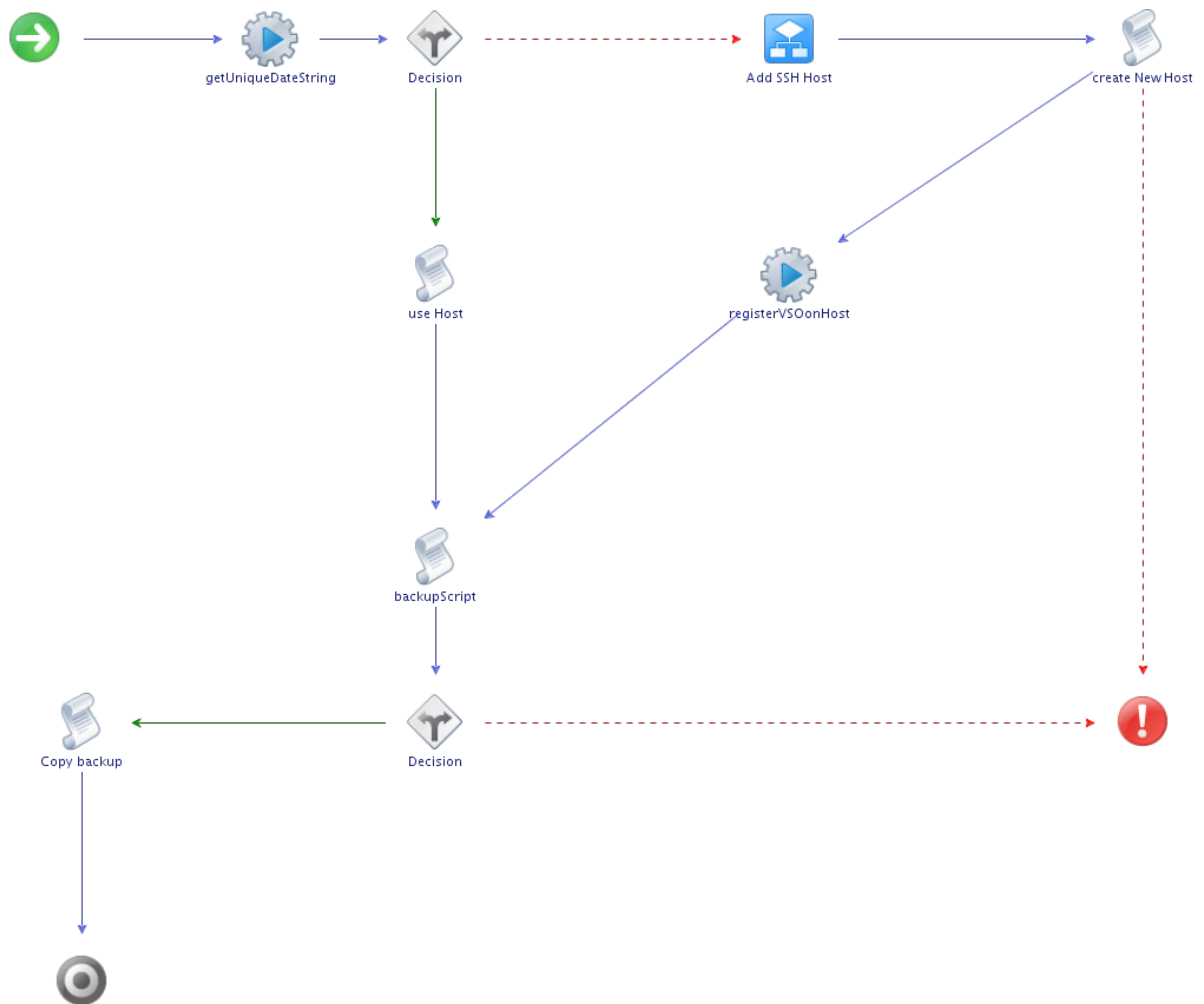
### 4. Parameter presentation

Parameter	Description	Parameter Info

Parameter	Description	Parameter Info
blnNewHost	Add a new vCenter Server Appliance SSH host?	Default value: false
sshNewHost	vCenter Server Appliance FQDN / IP	Mandatory input: #blnNewHost
sshNewPort	vCenter Server Appliance SSH Port	Default value: 22.0 Mandatory input: #blnNewHost
sshNewUsername	vCenter Server Appliance SSH username	Mandatory input: #blnNewHost
sshNewPassword	Vcenter Server Appliance SSH password	Mandatory input: #blnNewHost
sshNewRootPath	Vcenter Server Appliance SSH Root Path	
sshNewKeyPairPath	vCenter Server Appliance Key pair path	Default value: #keyPairPath Mandatory input: #blnNewHost
sshNewKeyPairPassword	vCenter Server Appliance Key pair password	
sshExistingHost	SSH Host that is registered	
sshExistingKeyPairPath	vCenter Server Appliance Key pair path	Default value: #keyPairPath
sshExistingKeyPairPassword	vCenter Server Appliance Key pair password	
vcsaPythonBackupScript	Vcenter Server Appliance python backup full path	Default value: /opt/backup/python/backup_lin.py

Parameter	Description	Parameter Info
vcsaBackupLocation	Enter the backup location	Default value: /storage/core/backups/
remoteHost	Remote Host FQDN/IP	
remoteUser	Remote user	
remotePassword	Remote Password	
remoteBackupPath	Remote backup server location	

### 5. Workflow schema



### 6. Workflow items

Name	Type	Next item	Description

Name	Type	Next item	Description
backupScript	WorkflowItemTask	Decision	
	WorkflowItemEnd		
Copy backup	WorkflowItemTask		
getUniqueDateString	WorkflowItemTask	Decision	Get a unique string based on current date with milliseconds
Decision	WorkflowItemGenericCondition	Copy backup	
	WorkflowItemEnd		
create New Host	WorkflowItemTask	registerVS0onHost	
registerVS0onHost	WorkflowItemTask	backupScript	Upload and register the VS-0 public key to the target user@host. This will enable future connections to use public key authentication instead of password authentication for this user.
Decision	WorkflowItemGenericCondition	use Host	
use Host	WorkflowItemTask	backupScript	
Add SSH Host	WorkflowItemLink	create New Host	Adds a new SSH Host to configuration

## 7. Source code for the used actions

### 7.1. backupScript

```
var session = null;

try {

    if(sshHost==null){
        errorHandler('SSH Host = null');
    }
}
```

```
        throw new Error('SSH Host = null');
    }

    System.log("Start Session _____");

    session = new SSHSession(sshHost.hostname, sshHost.username, sshHost.port);
    session.connectWithIdentity(keyPairPath, keyPairPassword);

    vcsaBackupFullpath = vcsaBackupLocation + '/' + strUniqueDate
+ '_' + sshHost.hostname + '.bak' ;

    System.log("Backup fullpath = '" + vcsaBackupFullpath + "'");
    var backupCommand = 'sudo '+vcsaPythonBackupScript + ' -f ' + vcsaBackupFullpath;
    System.log("Executing '" + backupCommand + "'");

    session.executeCommand(backupCommand, true);
    exitCode = session.exitCode;

} catch (e) {
    vcsaBackupFullpath='';
    throw "Error : " + e;

} finally {
    System.log("_____");
    if (session) {
        System.log("Output: '" + session.getOutput() + "'");
        System.log("Error: '" + session.getError() + "'");
        session.disconnect();
    }
    System.log("Exit code: '" + exitCode + "'");
}

function errorHandler(errorString){
    System.log("_____");
    System.error("An error occurred : '"+ errorString +"'")
    System.error("Exiting the script with error code -1" )
    exitCode = '-1';
    error="An error occurred : '"+ errorString +"'";
}
}
```

## 7.2. Copy backup

```
System.log("Login to SSH Host and SCP the file ");
var session = null;
try {
  if(sshHost==null){
    errorHandler('SSH Host = null');
    throw new Error('SSH Host = null');
  }

  System.log("Start SCP Copy Session");

  session = new SSHSession(sshHost.hostname, sshHost.username, sshHost.port);
  session.connectWithIdentity(keyPairPath,keyPairPassword);

  var scpCommand = 'sudo sshpass -p "' +remotePassword + '" scp ' + vcsaBackupFullpath+'
'+remoteUser+'@'+remoteHost+':'+remoteBackupPath;
  System.log('Executing sudo sshpass -p "*****" scp ' + vcsaBackupFullpath+'
'+remoteUser+'@'+remoteHost+':'+remoteBackupPath );
  session.executeCommand(scpCommand, true);
  exitCode = session.exitCode;

} catch (e) {
  throw "Error : " + e;

} finally {
  System.log("_____");
  if (session) {
    System.log("Output: '" + session.getOutput() + "'");
    System.log("Error: '" + session.getError() + "'");
    session.disconnect();
  }
  System.log("Exit code: '" + exitCode + "'");
}

function errorHandler(errorString){
  System.log("_____");
  System.error("An error occurred : '"+ errorString +"'")
  System.error("Exiting the script with error code -1" )
  exitCode = '-1';
  error="An error occurred : '"+ errorString +"'";
}
```

### 7.3. getUniqueDateString

```
//Auto generated script, cannot be modified !  
actionResult = System.getModule("nl.lumc.library.utils").getUniqueDateString() ;
```

### 7.4. Decision

```
//Generated by the system, cannot be edited  
return (exitCode == "0") ;
```

### 7.5. create New Host

```
System.log("Using new host");  
keyPairPath=sshNewKeyPairPath;  
keyPairPassword=sshNewKeyPairPassword;
```

### 7.6. registerVS0onHost

```
//Auto generated script, cannot be modified !  
System.getModule("com.vmware.library.ssh").registerVS0onHost(host,username,password,localPath,remotePath) ;
```

### 7.7. Decision

```
//Generated by the system, cannot be edited  
return (blnNewHost == false) ;
```

### 7.8. use Host

```
if(! blnNewHost){  
    System.log("Using existing host");  
    sshHost=sshExistingHost;  
    keyPairPath=sshExistingKeyPairPath;  
    keyPairPassword=sshExistingKeyPairPassword;  
}
```