



# 使用Ansible实现NetApp SAP Landscape Management集成

## NetApp solutions for SAP

NetApp  
October 30, 2025

# 目录

使用Ansible实现NetApp SAP Landscape Management集成	1
TR-4953: 《使用Ansible实现NetApp SAP Landscape Management集成》	1
SAP系统克隆、复制和刷新场景	1
系统刷新、复制和克隆的用例	2
解决逻辑损坏问题	3
灾难恢复测试	4
使用Ansible实现NetApp SAP Lama集成	4
实施示例	5
经验证的配置和限制	5
实验室设置	6
SAP Lama配置	7
SAP Lama配置 workflow—克隆系统	10
SAP Lama取消配置 workflow—系统销毁	18
SAP Lama配置 workflow—复制系统	21
SAP Lama配置 workflow—系统刷新	25
提供程序脚本配置和Ansible攻略手册	27
提供程序配置文件netapp_clone.conf	28
提供程序脚本netapp_clone.sh	28
Ansible攻略手册NetApp_Lam_CloneVolumes.yml	36
Ansible攻略手册NetApp_Lam_ServiceConfigRemove.yml	37
Ansible攻略手册NetApp_Lama_ClearMountConfig.yml	38
Ansible inventory.yml示例	39
结论	40
从何处查找追加信息	40
版本历史记录	41

# 使用Ansible实现NetApp SAP Landscape Management集成

## TR-4953: 《使用Ansible实现NetApp SAP Landscape Management集成》

SAP Landscape Management (Lama)支持SAP系统管理员自动执行SAP系统操作、包括端到端SAP系统克隆、复制和刷新操作。

作者: Michael Schlerser、Nils Bauer、NetApp

NetApp提供了一组丰富的Ansible模块、SAP Lama可以通过SAP Lama Automation Studio访问NetApp Snapshot和FlexClone等技术。这些技术有助于简化和加速SAP系统克隆、复制和刷新操作。

在内部运行NetApp存储解决方案的客户或在Amazon Web Services、Microsoft Azure或Google Cloud Platform等公共云提供商中使用NetApp存储服务的客户均可使用此集成。

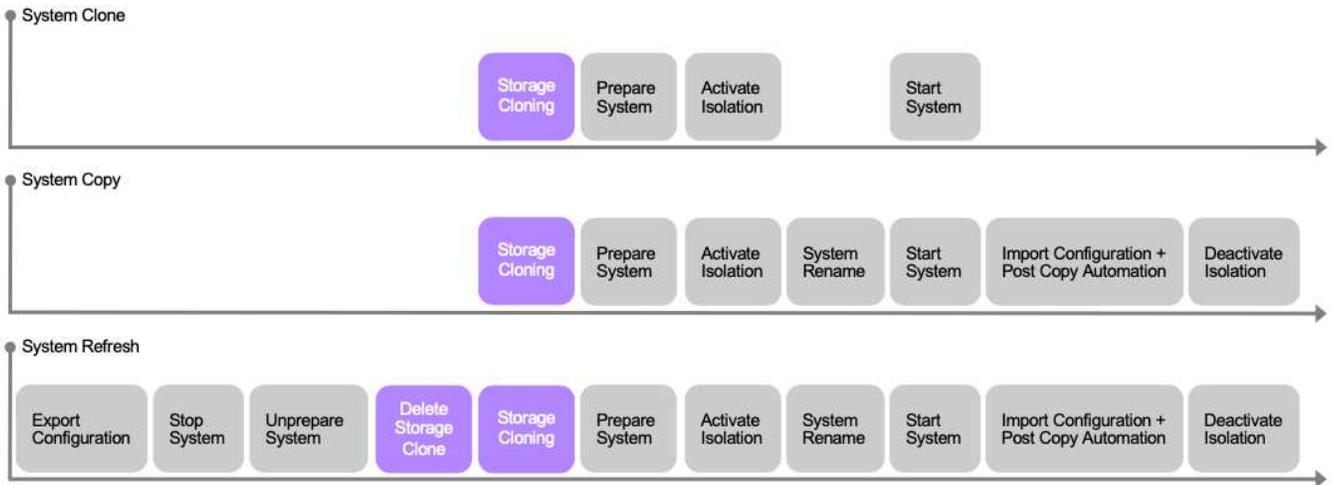
本文档介绍了如何为SAP Lama配置NetApp存储功能、以便使用Ansible自动化执行SAP系统复制、克隆和刷新操作。

## SAP系统克隆、复制和刷新场景

术语SAP系统副本通常用作三个不同流程的同义词: SAP系统克隆、SAP系统副本或SAP系统刷新。请务必区分不同的操作、因为每个操作的工作流和用例都不同。

- \* SAP系统克隆。\* SAP系统克隆是源SAP系统的相同克隆。SAP系统克隆通常用于解决逻辑损坏或测试灾难恢复场景。在执行系统克隆操作时、主机名、实例编号和SID保持不变。因此、必须为目标系统建立适当的网络隔离、以确保与生产环境之间不存在通信。
- \* SAP系统副本。\* SAP系统副本是使用源SAP系统中的数据设置的新目标SAP系统。例如、新的目标系统可以是一个包含生产系统数据的附加测试系统。源系统和目标系统的主机名、实例编号和SID不同。
- \* SAP系统刷新。\* SAP系统刷新是使用源SAP系统中的数据刷新现有目标SAP系统。目标系统通常是SAP传输环境的一部分、例如质量保证系统、该系统会使用生产系统中的数据进行更新。源系统和目标系统的主机名、实例编号和SID不同。

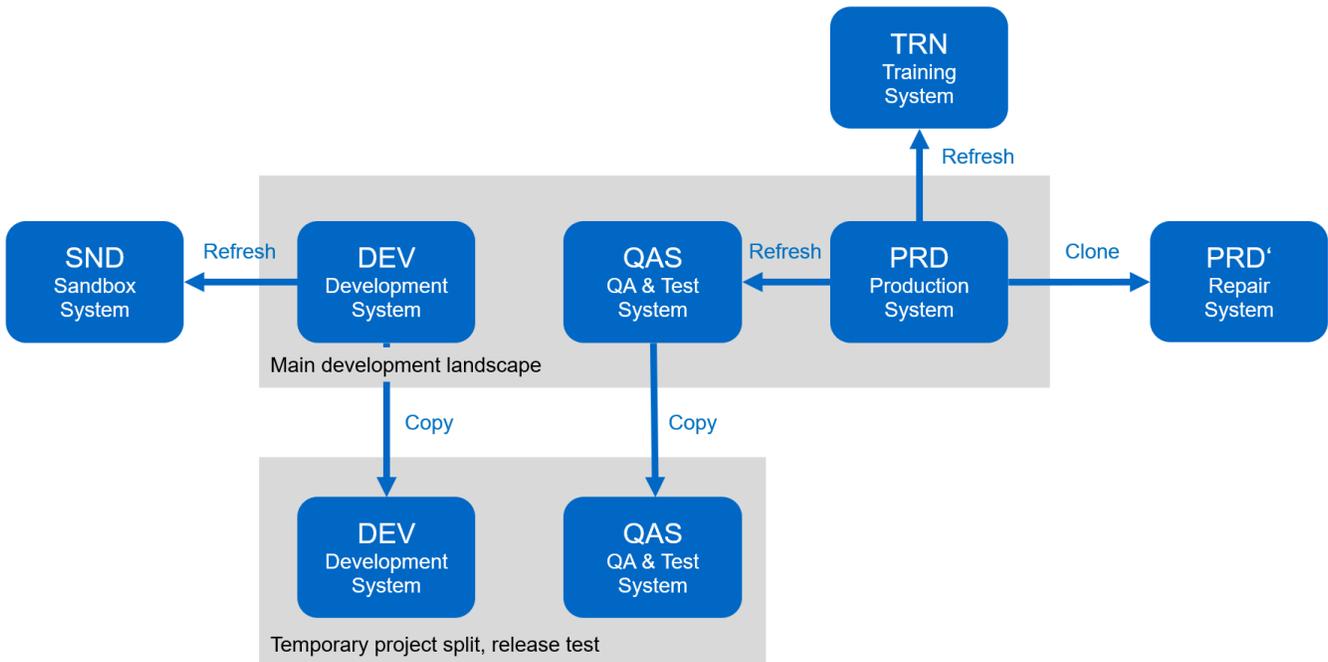
下图说明了在执行系统克隆、系统副本或系统刷新操作期间必须执行的主要步骤。紫色框表示可集成NetApp存储功能的步骤。所有这三项操作均可通过SAP Lama实现完全自动化。



## 系统刷新、复制和克隆的用例

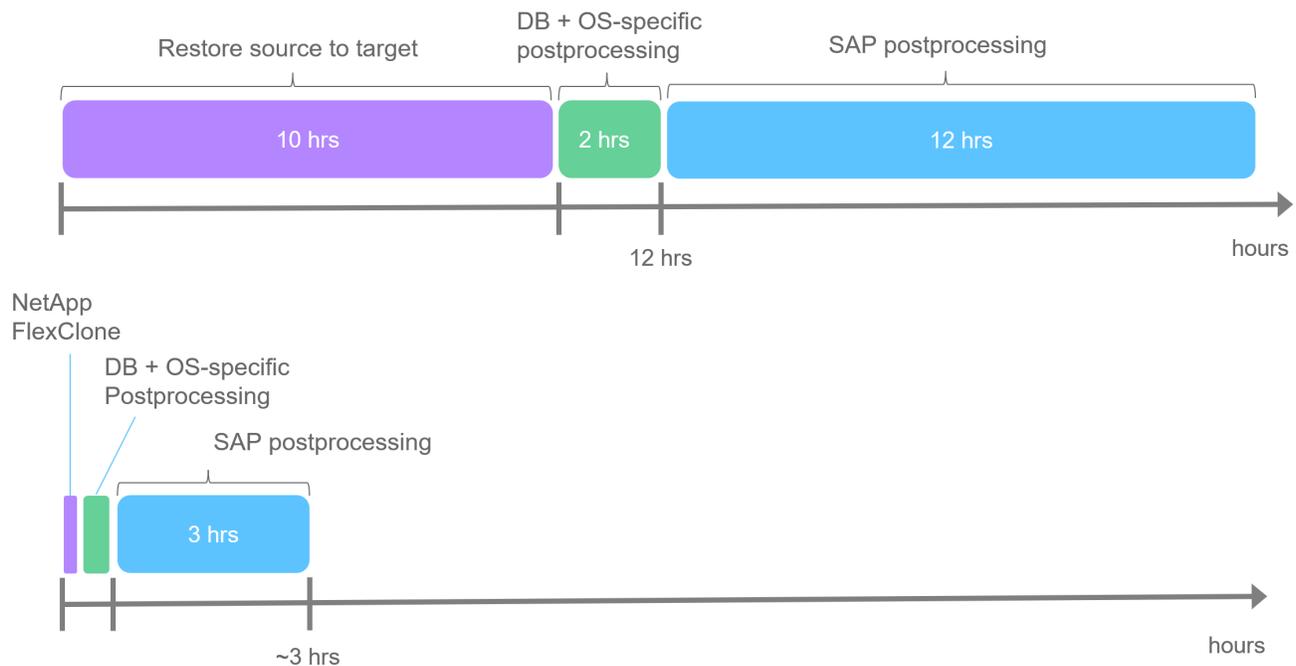
在多种情况下、必须将源系统中的数据提供给目标系统、以便用于测试或培训目的。必须定期使用源系统中的数据更新这些测试和培训系统、以确保使用当前数据集执行测试和培训。

这些系统刷新操作包括基础架构层、数据库层和应用程序层上的多个任务、它们可能需要数天时间、具体取决于自动化级别。



SAP Lama和NetApp克隆 workflow 可用于加速和自动化基础架构和数据库层的所需任务。SAP Lama不会将备份从源系统还原到目标系统、而是使用NetApp Snapshot副本和NetApp FlexClone技术、以便在几分钟内即可完成启

动的HANA数据库所需的任务、而不是像下图所示的几小时内完成。克隆过程所需的时间与数据库大小无关；因此、即使是非常大的系统也只需几分钟即可创建完毕。通过自动执行操作系统和数据库层以及SAP后处理端的任务、可以进一步缩短运行时间。



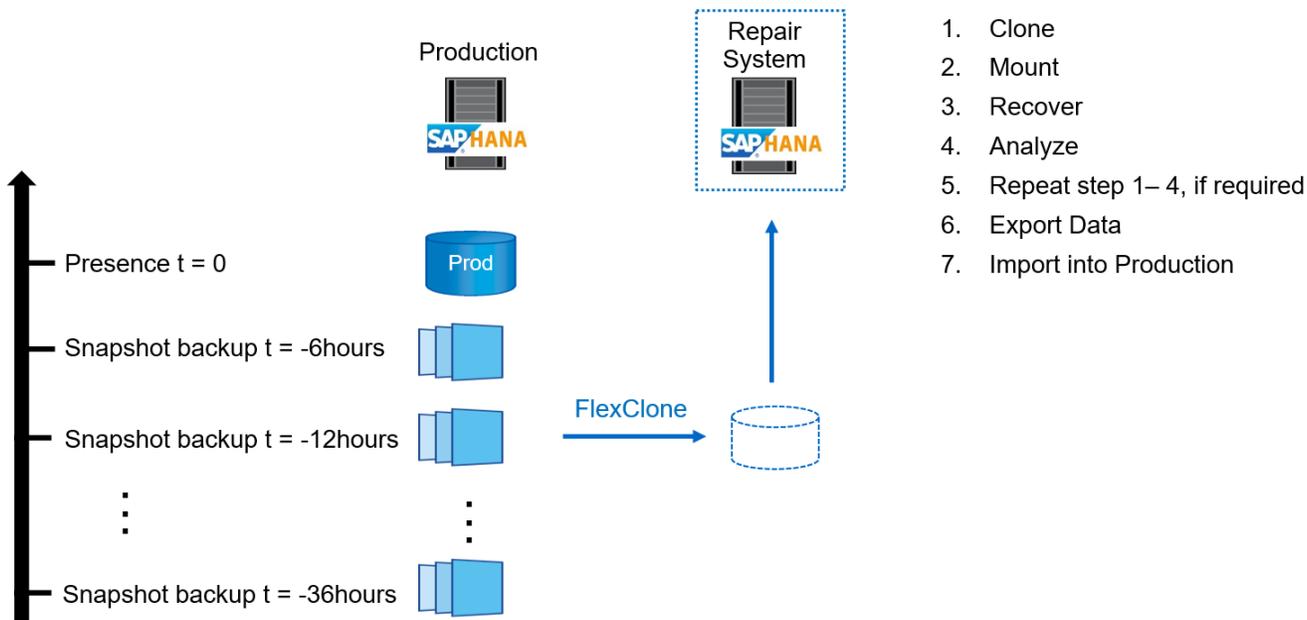
## 解决逻辑损坏问题

逻辑损坏可能是由软件错误，人为错误或破坏引起的。遗憾的是，使用标准高可用性和灾难恢复解决方案往往无法解决逻辑损坏问题。因此、根据发生逻辑损坏的层、应用程序、文件系统或存储、有时可能无法满足最短停机时间和可接受的数据丢失要求。

最糟糕的情况是SAP应用程序中的逻辑损坏。SAP 应用程序通常在不同应用程序相互通信并交换数据的环境中运行。因此，不建议使用还原和恢复发生逻辑损坏的 SAP 系统。将系统还原到损坏发生前的某个时间点会导致数据丢失。此外，SAP 环境将不再同步，需要额外的后处理。

更好的方法是尝试通过在单独的修复系统中分析问题来修复系统中的逻辑错误、而不是还原SAP系统。根发生原因分析需要业务流程和应用程序所有者的参与。在这种情况下，您可以根据发生逻辑损坏之前存储的数据创建修复系统（生产系统的克隆）。在修复系统中、可以将所需数据导出并导入到生产系统中。通过这种方法、生产系统无需停止、在最佳情况下、不会丢失任何数据、也不会丢失一小部分数据。

设置修复系统时、灵活性和速度至关重要。借助基于NetApp存储的Snapshot备份、可以使用NetApp FlexClone 技术使用多个一致的数据映像来创建生产系统的克隆。如果使用基于文件的备份的重定向原来设置修复系统、则只需几秒钟即可创建FlexClone卷、而不是多小时。

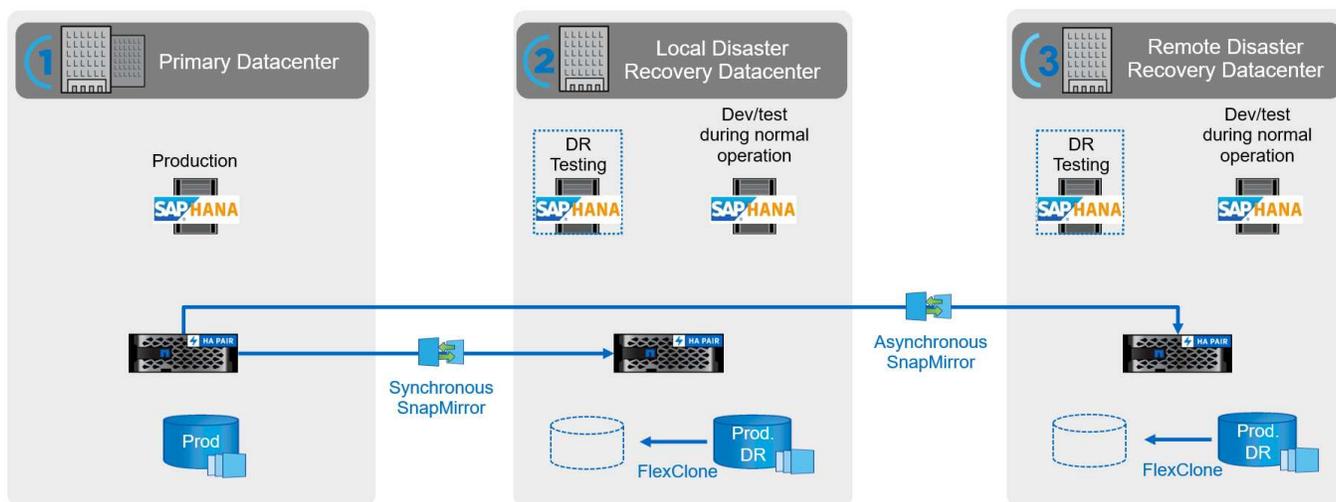


## 灾难恢复测试

有效的灾难恢复策略需要测试所需的工作流。测试可证明该策略是否有效、以及内部文档是否足以满足要求。此外、管理员还可以通过它对所需的过程进行培训。

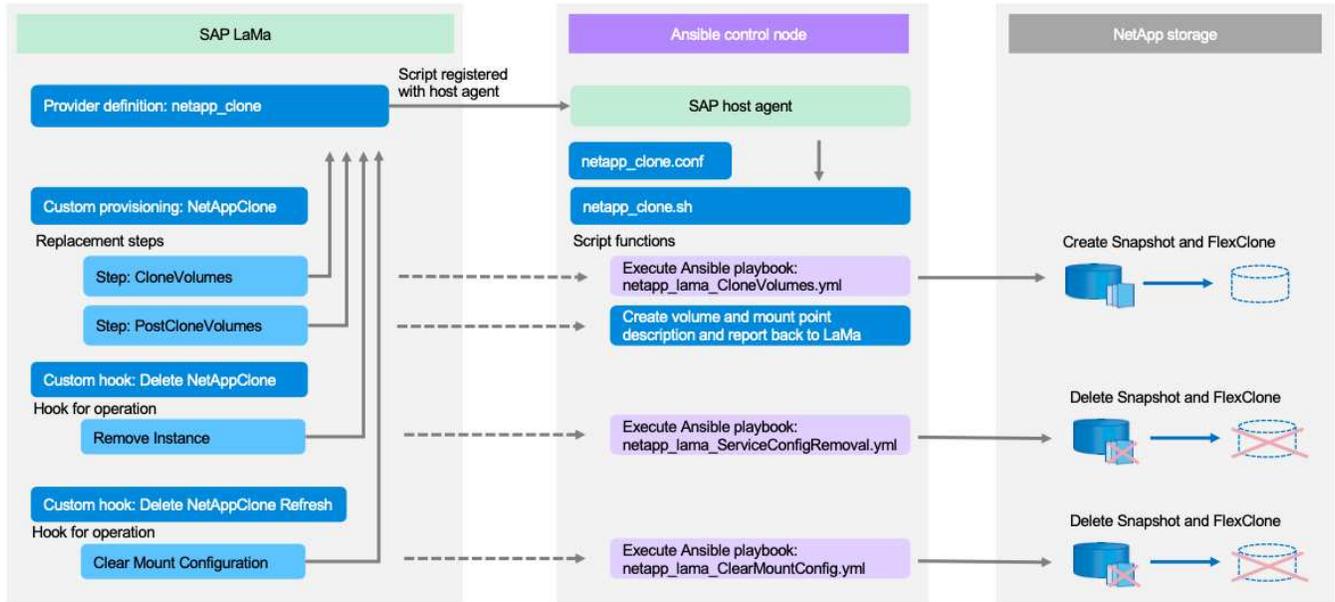
使用SnapMirror进行存储复制可以执行灾难恢复测试、而不会使RTO和RPO面临风险。可以在不中断数据复制的情况下执行灾难恢复测试。异步和同步SnapMirror的灾难恢复测试会在灾难恢复目标上使用Snapshot备份和FlexClone卷。

SAP Lama可用于编排整个测试操作步骤、还负责网络隔离、目标主机维护等。



## 使用Ansible实现NetApp SAP Lama集成

此集成方法将SAP Lama自定义配置和操作挂钩与Ansible攻略手册结合使用、用于NetApp存储管理。下图简要概述了Lama端的配置以及示例实施的相应组件。



充当Ansible控制节点的中央主机用于执行SAP Lama发出的请求、并使用Ansible攻略手册触发NetApp存储操作。必须在此主机上安装SAP主机代理组件、才能将此主机用作与SAP Lama的通信网关。

在Lama Automation Studio中、定义了一个在Ansible主机的SAP主机代理上注册的提供程序。主机代理配置文件指向一个Shell脚本、此脚本由SAP Lama使用一组命令行参数调用、具体取决于请求的操作。

在Lama Automation Studio中、定义了自定义配置和自定义连接、以便在配置期间以及在系统取消配置后的清理操作期间执行存储克隆操作。然后、Ansible控制节点上的shell脚本将执行相应的Ansible攻略手册、从而触发Snapshot和FlexClone操作、并通过取消配置 workflows 删除克隆。

有关NetApp Ansible模块和Lama提供程序定义的详细信息、请访问：

- "NetApp Ansible模块"
- "SAP Lama文档—提供程序定义"

## 实施示例

由于系统和存储设置具有大量可用选项、因此应使用示例实施作为您的各个系统设置和配置要求的模板。



示例脚本按原样提供、NetApp不支持。您可以通过电子邮件发送电子邮件至：<mailto:ng-sapcc@netapp.com> [ng-sapcc@netapp.com](mailto:ng-sapcc@netapp.com)^请求最新版本的脚本。

### 经验证的配置和限制

以下原则适用于示例实施、可能需要进行调整以满足客户需求：

- 受管SAP系统使用NFS访问NetApp存储卷、并根据自适应设计原则进行设置。
- 您可以使用NetApp Ansible模块(ZAPI和REST API)支持的所有ONTAP 版本。

- 单个NetApp集群和SVM的凭据在提供程序脚本中硬编码为变量。
- 存储克隆是在源SAP系统使用的同一存储系统上执行的。
- 目标SAP系统的存储卷与附录中的源卷名称相同。
- 未在二级存储(SV/SM)上实施克隆。
- 未实施FlexClone拆分。
- 源SAP系统和目标SAP系统的实例编号相同。

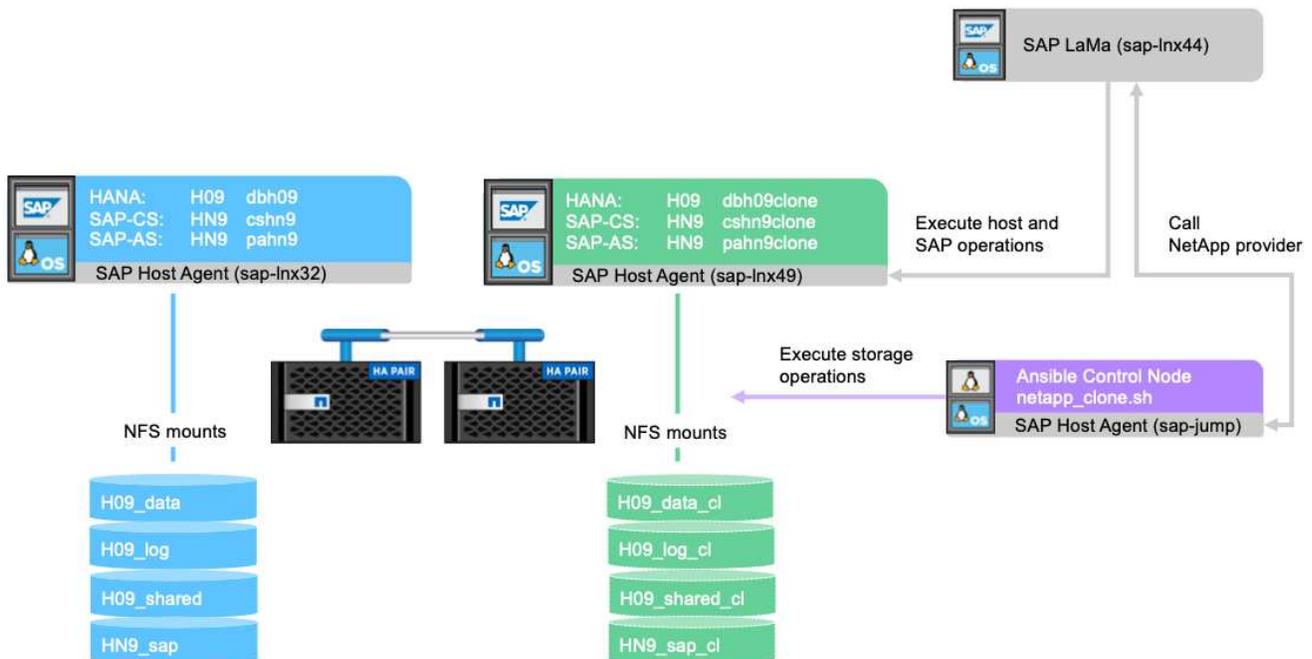
## 实验室设置

下图显示了我们使用的实验室设置。用于系统克隆操作的源SAP系统HN9由数据库H09、SAP CS和SAP作为服务运行在安装了同一主机(sap-lnx32)上 "自适应设计" 已启用。根据准备了一个Ansible控制节点 "适用于NetApp ONTAP 的Ansible攻略手册" 文档。

此主机上也安装了SAP主机代理。NetApp提供程序脚本和Ansible游戏攻略手册已在Ansible控制节点上配置、如中所述 "附录：提供程序脚本配置。"

主机 sap-lnx49 用作SAP Lama克隆操作的目标、并在此配置了隔离就绪功能。

系统复制和刷新操作使用不同的SAP系统(HNA作为源、HN2作为目标)、因为在该系统上启用了后复制自动化(PCA)。



实验室设置中使用了以下软件版本：

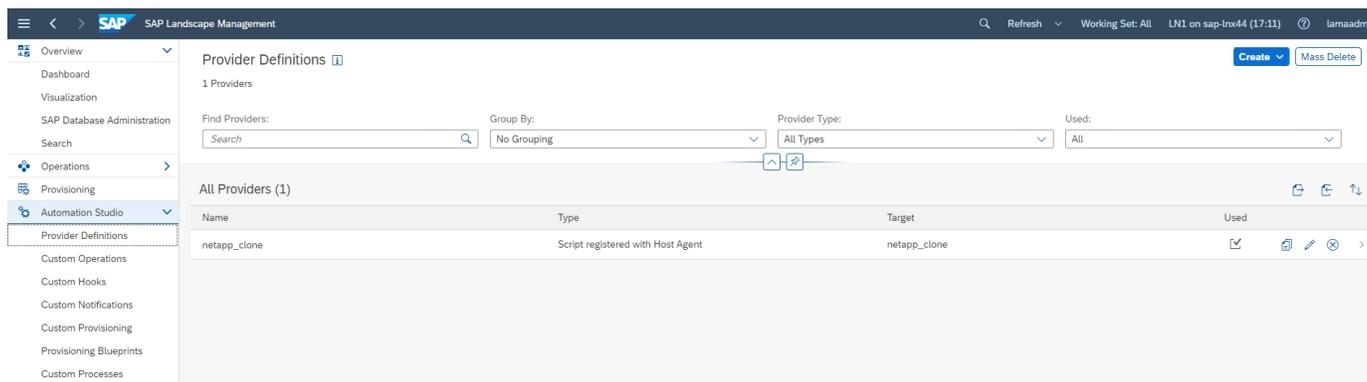
- SAP LaMa Enterprise Edition 3.00 SP23\_2
- SAP HANA 2.00.052.00.1599235305
- SAP 7.77修补程序27 (S/4 HANA 1909)

- SAP Host Agent 7.22修补程序56
- SAPCACEXT 7.22修补程序69
- Linux SLES 15 SP2
- Ansible 2.13.7.
- NetApp ONTAP 9.8P8

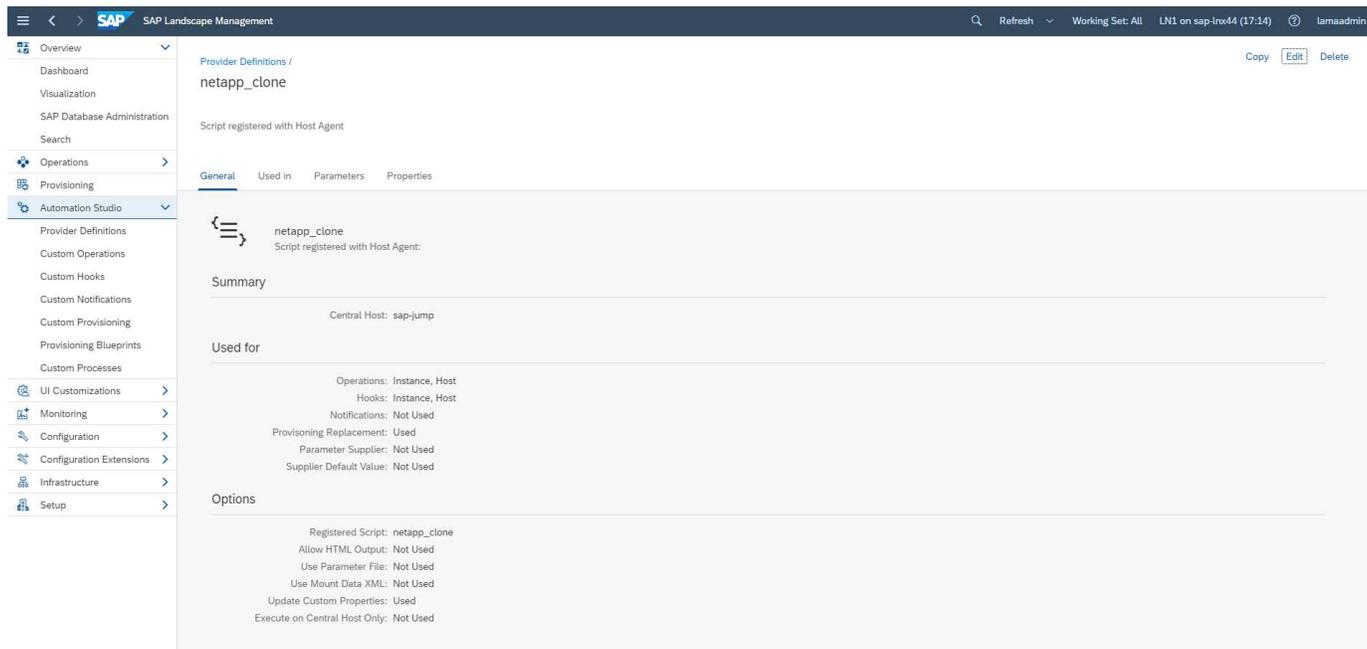
## SAP Lama配置

### SAP Lama提供程序定义

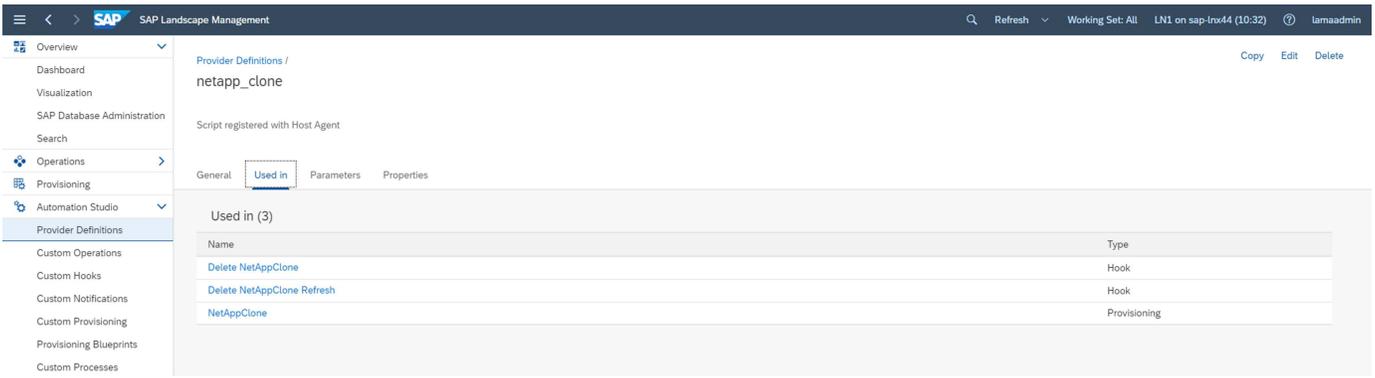
提供程序定义在SAP LaMa的Automation Studio中执行、如以下屏幕截图所示。示例实施使用一个提供程序定义、该定义用于不同的自定义配置步骤和操作挂钩、如前面所述。



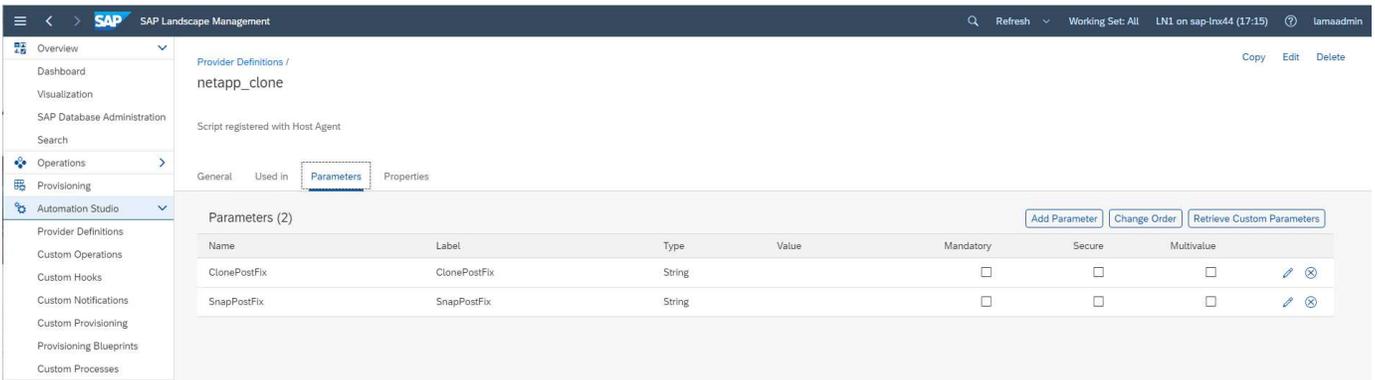
提供程序 netapp\_clone 定义为脚本 netapp\_clone.sh 已在SAP主机代理上注册。SAP主机代理在中央主机上运行 sap-jump、也充当Ansible控制节点。



已用英寸\*选项卡显示了此提供程序用于哪些自定义操作。以下章节将显示自定义配置 NetApp克隆\*和自定义挂钩\*删除NetApp克隆\*和\*删除NetApp克隆刷新\*的配置。

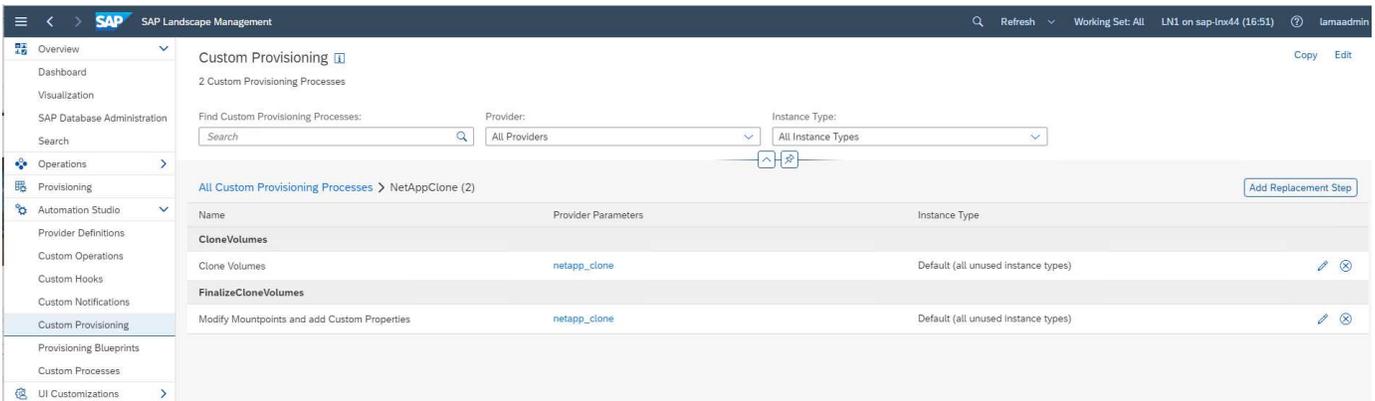


在执行配置工作流期间会请求参数\*克隆后修复\*和\* Snap后 修复\*、这些参数用于Snapshot和FlexClone卷名称。



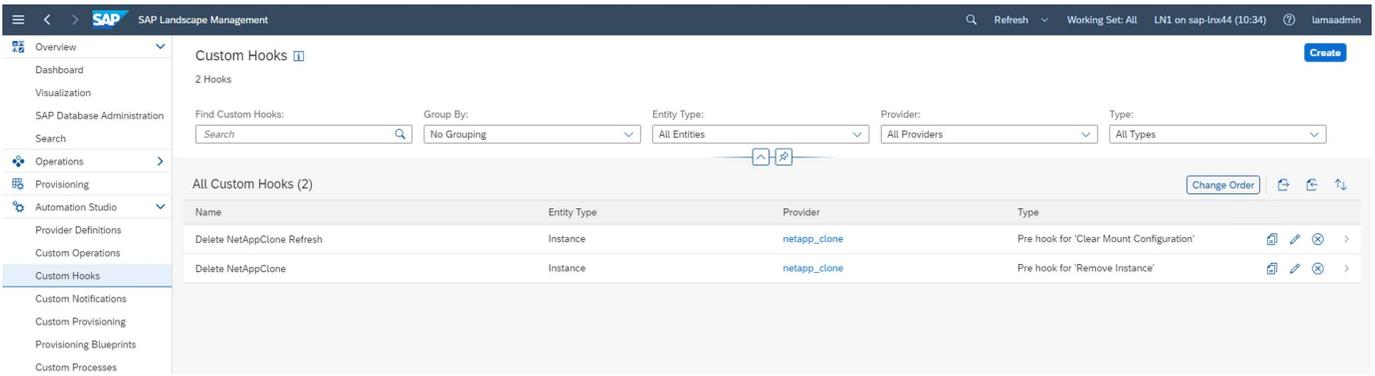
## SAP Lama自定义配置

在SAP Lama自定义配置中、使用前面所述的客户提供程序替换配置工作流步骤\*克隆卷\*和\*后克隆卷\*。

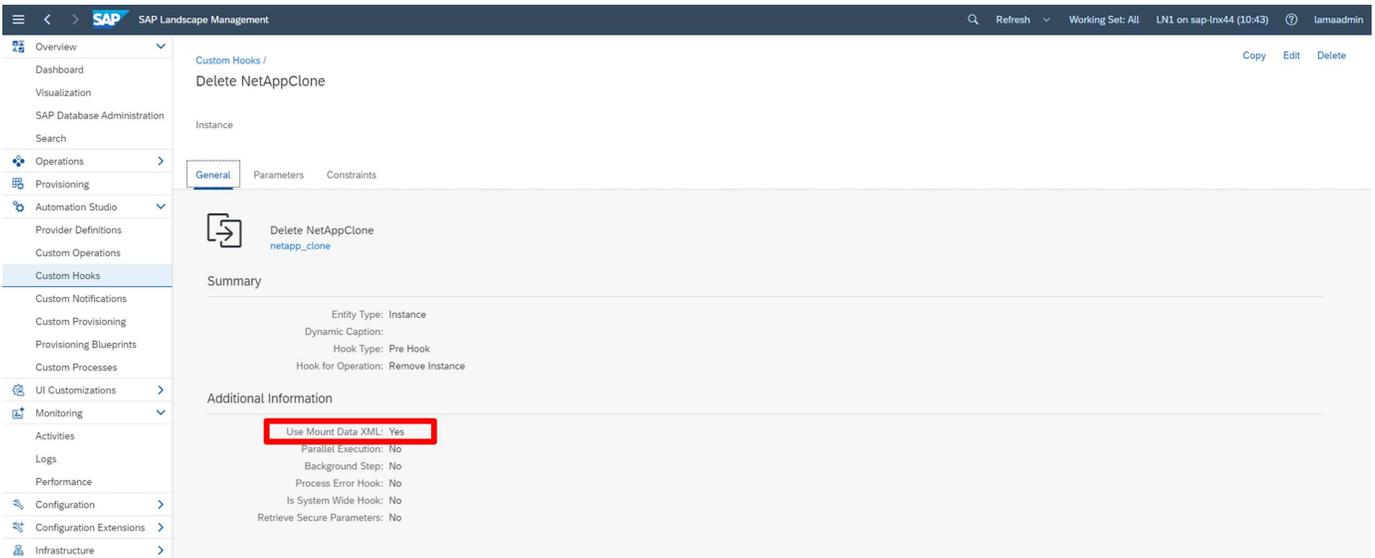


## SAP Lama自定义挂钩

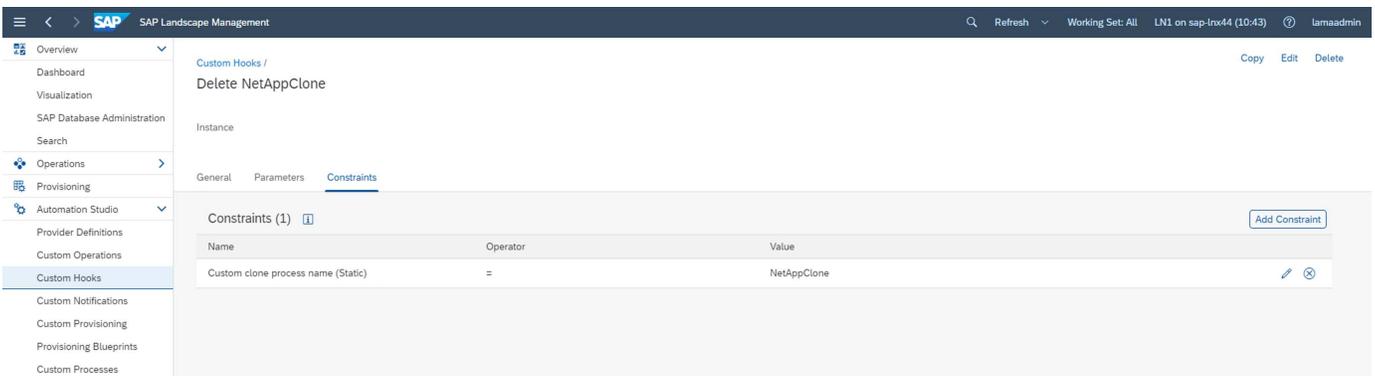
如果使用系统销毁工作流删除系统、则会使用hook \*删除NetAppClone\*调用提供程序定义 netapp\_clone。系统刷新工作流期间会使用\*删除NetApp克隆刷新\*挂钩、因为执行期间会保留此实例。



请务必为自定义挂钩配置\*使用挂载数据XML\*、以便SAP Lama向提供程序提供挂载点配置的信息。



为了确保只有在使用自定义配置 workflow 创建系统时才使用和执行自定义挂钩、系统中会添加以下限制。



有关使用自定义挂钩的详细信息、请参见 "SAP Lama文档"。

## 为SAP源系统启用自定义配置 workflow

要为源系统启用自定义配置 workflow、必须在配置中对其进行调整。必须选中具有相应自定义配置定义的\*使用自定义配置过程\*复选框。

SAP Landscape Management

Working Set: <AB> Search: [ ] Go LN1 on sap-lex4

Automation Studio Configuration Infrastructure

Pools Systems Hosts Characteristics

Overview of Systems and Instances

Discover Remove Instance and System Reassign Instances Mass Configuration Filtering Export Import

Name	Managed	AC-Enabled	Operational	Pool	Network	Description
HN9: NetWeaver ABAP 7.77, cshn9	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MUCCBC		
• System database: MASTER (configured) : H09, SAP HANA 02, dbh09	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MUCCBC	MUCCBC-SAP-Front	
• Central services: 01, cshn9	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MUCCBC	MUCCBC-SAP-Front	
• AS instance: 00, pahn9	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MUCCBC	MUCCBC-SAP-Front	
▶ HNA: NetWeaver ABAP 7.77, cshna	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUCCBC		

Systems: 2 Selected: HN9: NetWeaver ABAP 7.77, cshn9

System Details Log

Edit Show In

General

System Name: HN9: NetWeaver ABAP 7.77, cshn9

SID: HN9

Instance ID: SystemID: HN9, SystemHost: cshn9

Solution Manager settings

Assign Solution Manager System:

Focused Run Settings

Assign Focused Run System:

Disable Workmode Management:

System and AS Provisioning

This system was provided by:

This system can be used for:

Installation

Cloning  Application Server (Un-Installation)

Copying  Diagnostic Agent (Un-Installation)

Renaming  nZDM Java

Standalone PCA  Replication Configuration

Use Custom Provisioning Process:

NetAppClone

Use as TDMS Control System:

Is B/W Source System:

Use Replication for Single Tenant Database Refresh:

Intersystem Dependencies

From Instance	To Instance
• Outgoing (0)	
• Incoming (0)	

Entity Relations

Custom Relation Type	Target Entity Type	Target Entity
[i] Table is empty		

E-Mail Notification

Enable Email Notification:

Custom Notification

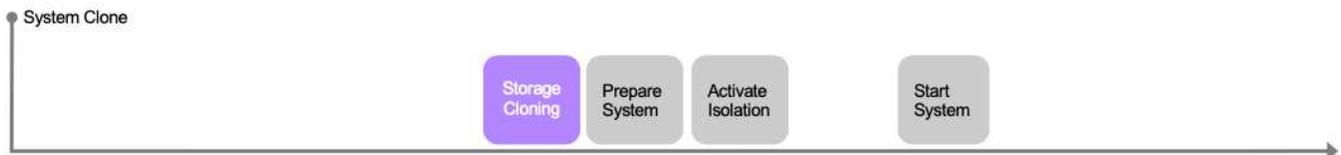
Enable Custom Notification:

ACM Settings

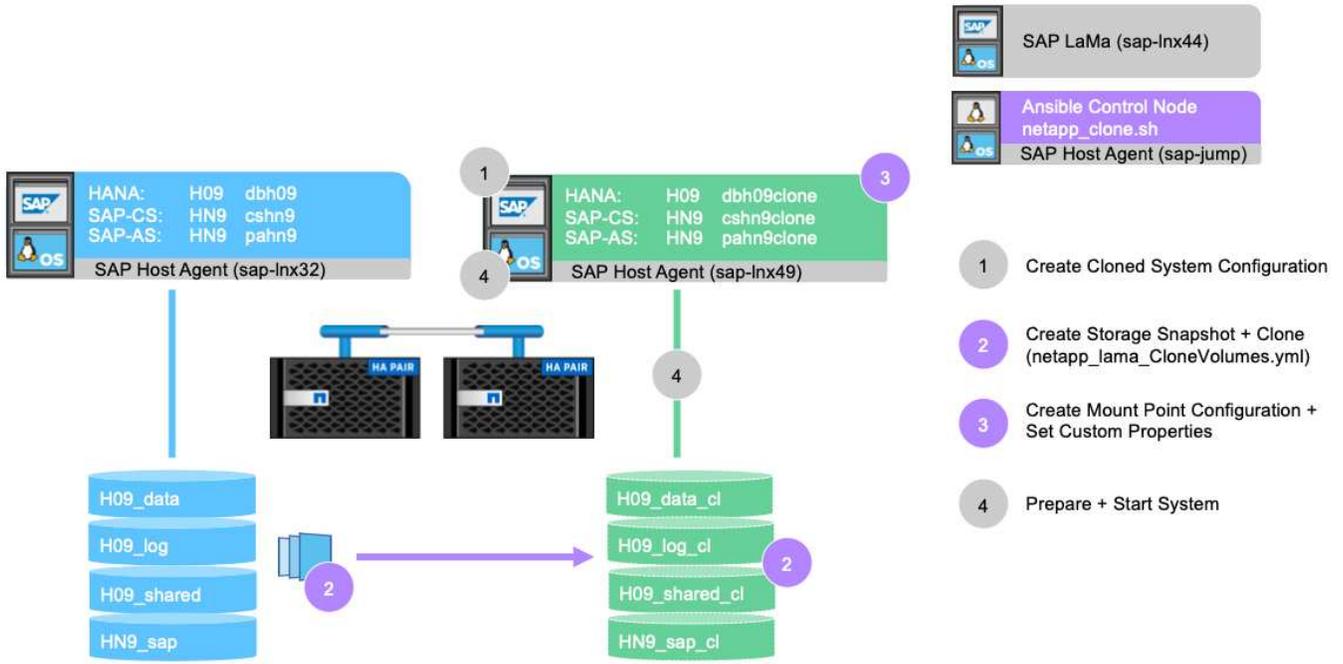
ACM-Managed:

## SAP Lama配置 workflow—克隆系统

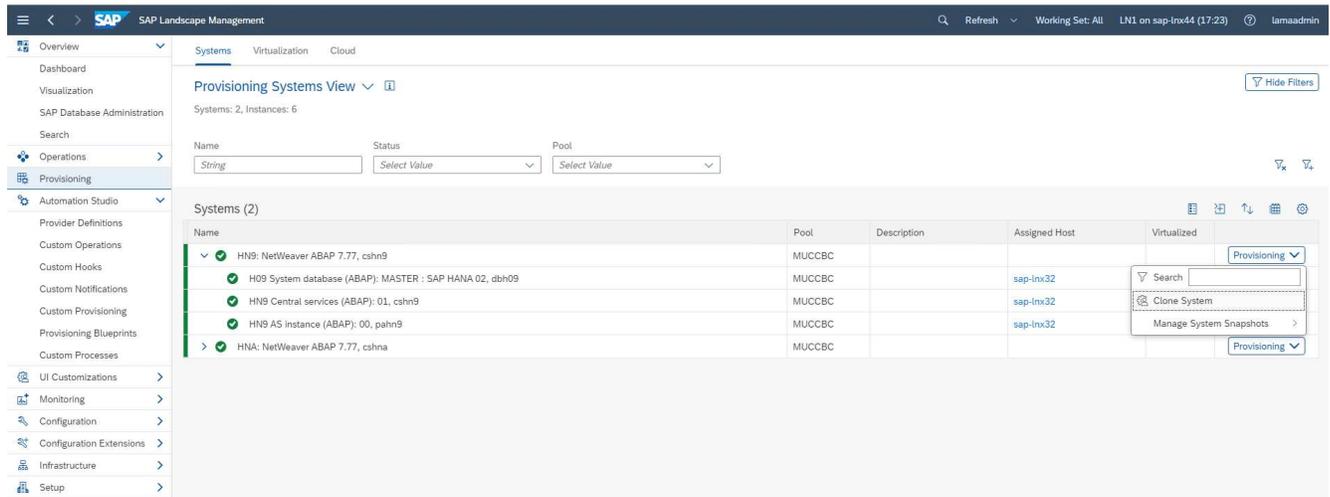
下图突出显示了使用系统克隆 workflow 执行的主要步骤。



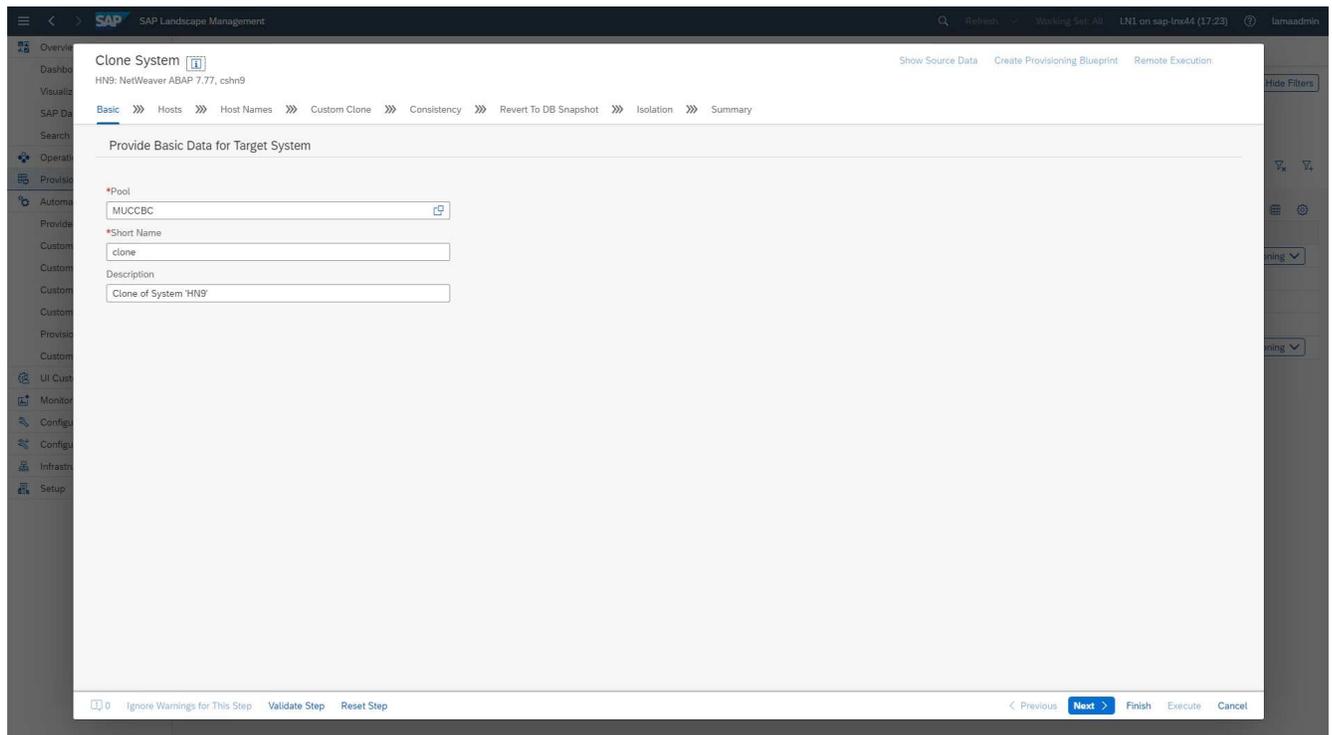
在本节中、我们将根据采用HANA数据库H09的源SAP系统HN9完成完整的SAP Lama系统克隆 workflow。下图概述了 workflow 期间执行的步骤。



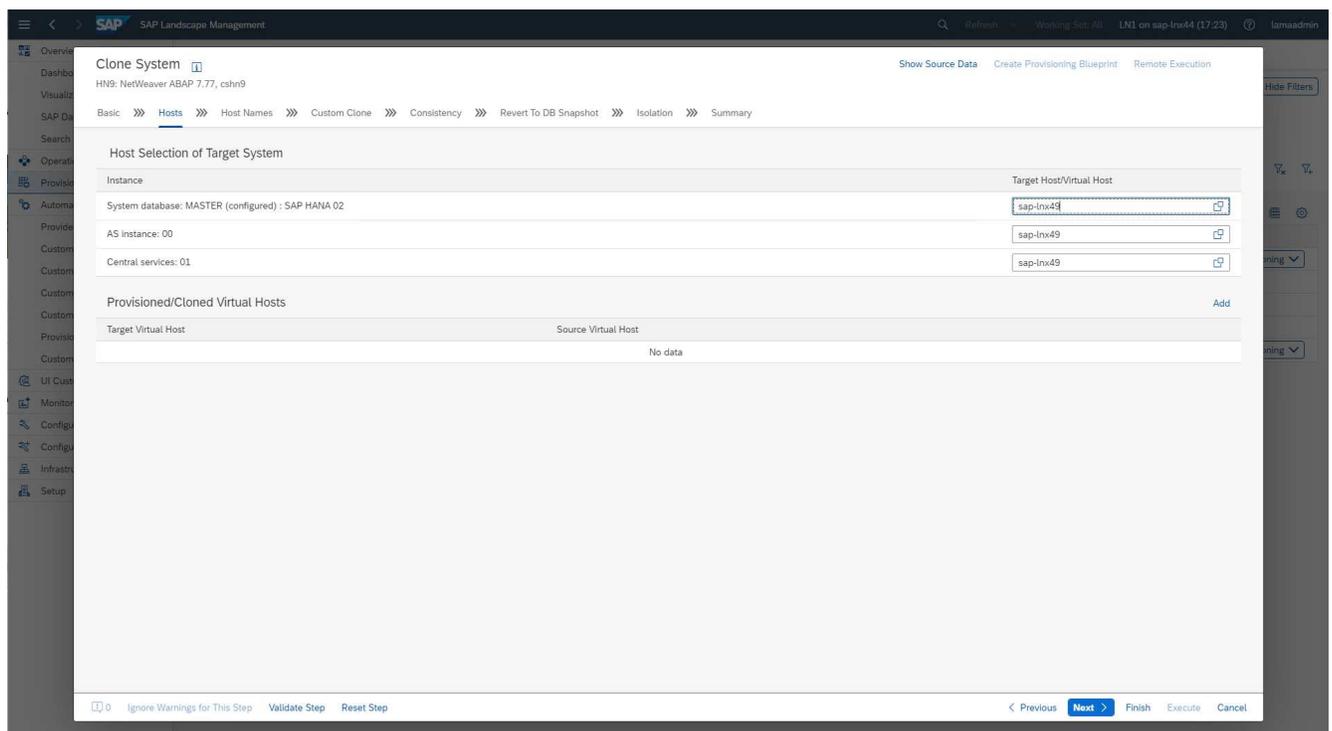
1. 要启动克隆 workflow、请打开菜单树中的\*配置\*、然后选择源系统(在我们的示例HN9中)。然后启动\*克隆系统\*向导。



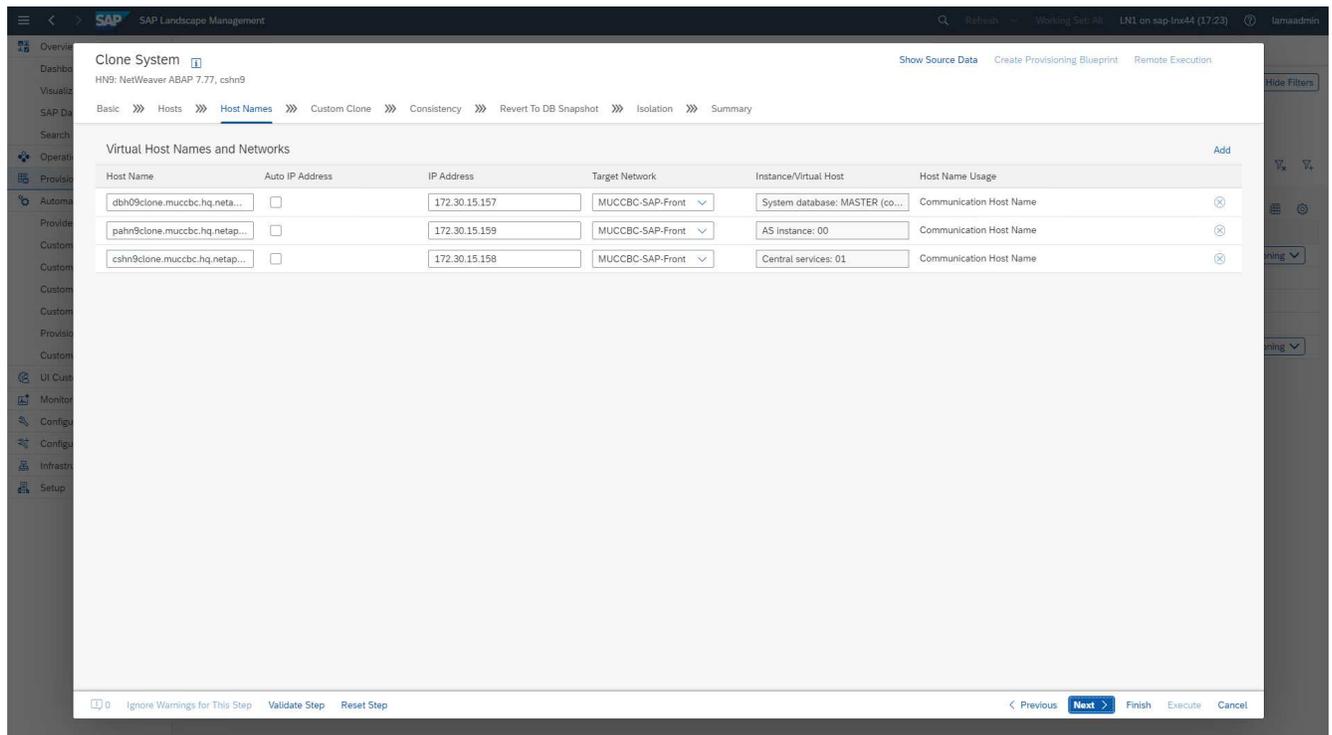
2. 输入请求的值。向导的屏幕1会要求您输入克隆系统的池名称。此步骤指定要启动克隆系统的实例(虚拟或物理)。默认情况下、将系统克隆到与目标系统相同的池中。



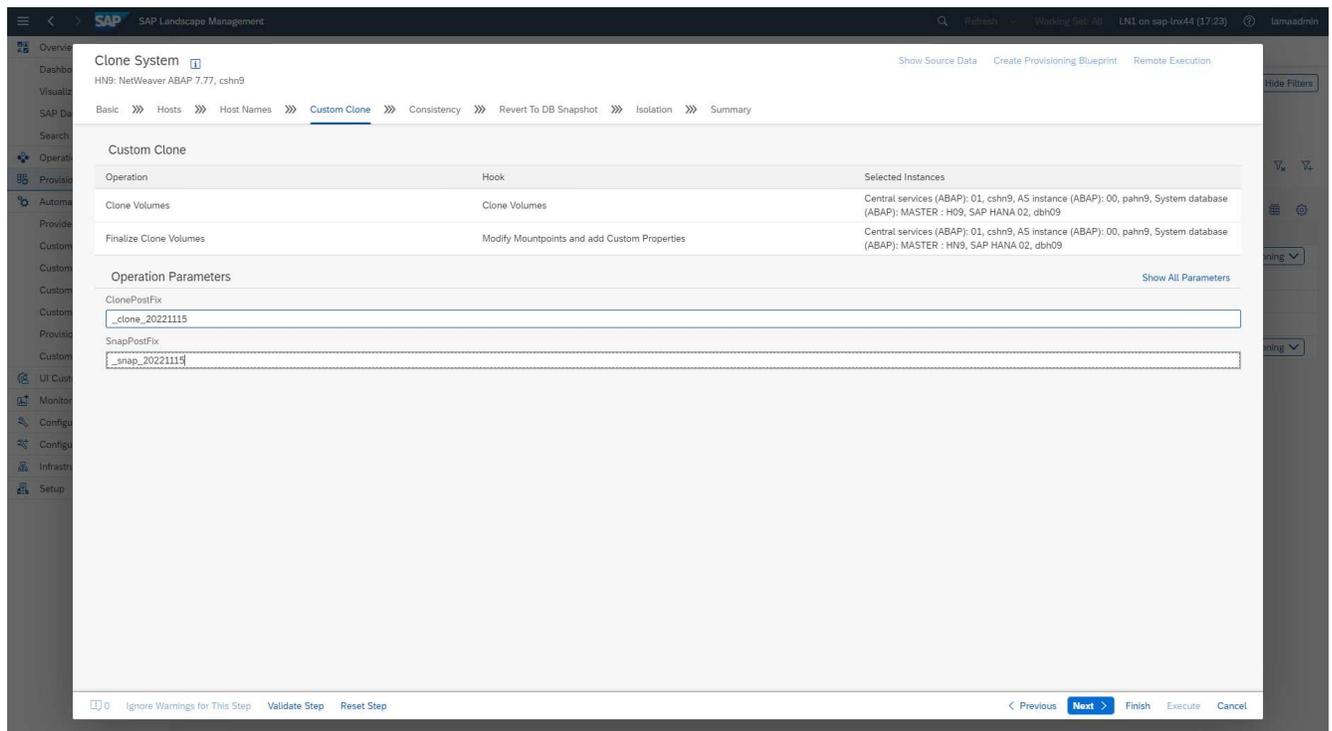
3. 此向导的屏幕2会询问启动新SAP实例的目标主机。可以从上一屏幕指定的主机池中选择此实例的目标主机。每个实例或服务都可以在不同的主机上启动。在我们的示例中、所有这三个服务都在同一主机上运行。



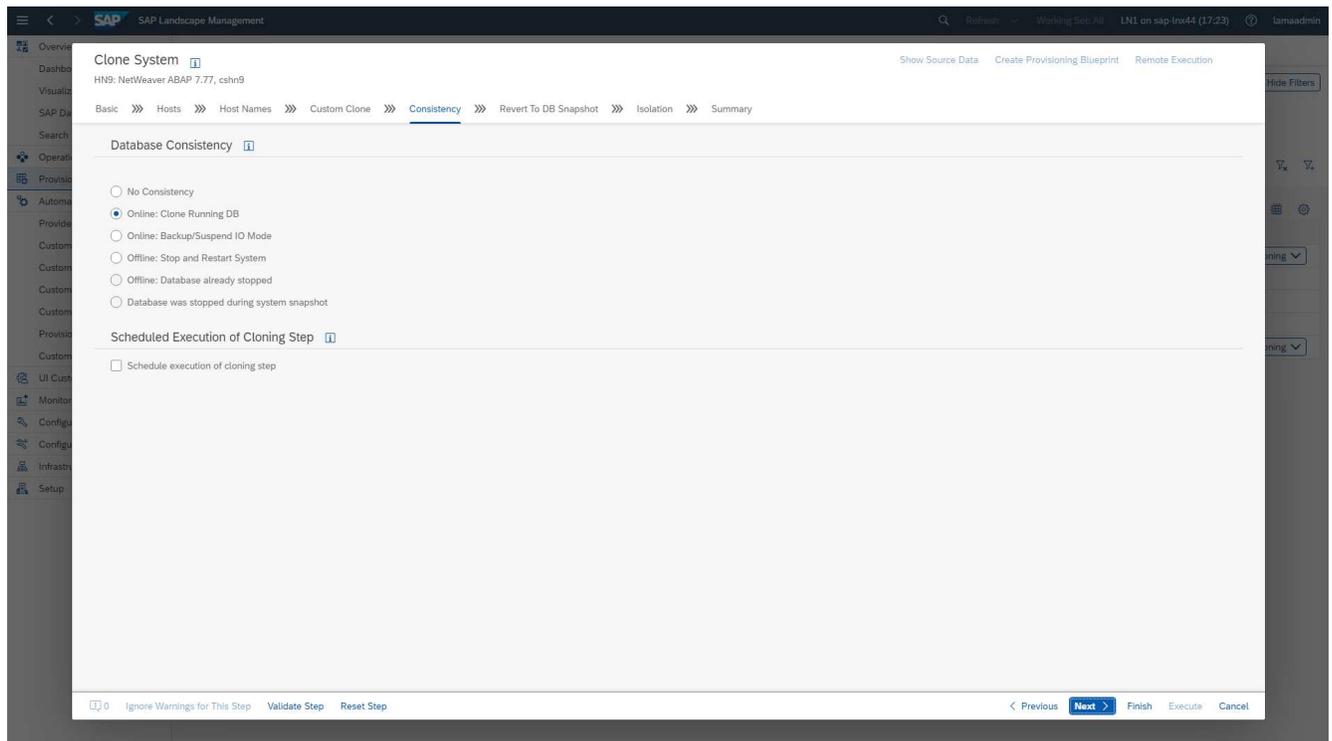
4. 在屏幕3中提供所需的信息、该屏幕要求提供虚拟主机名和网络。通常、主机名会保留在DNS中、因此IP地址会相应地预先填充。



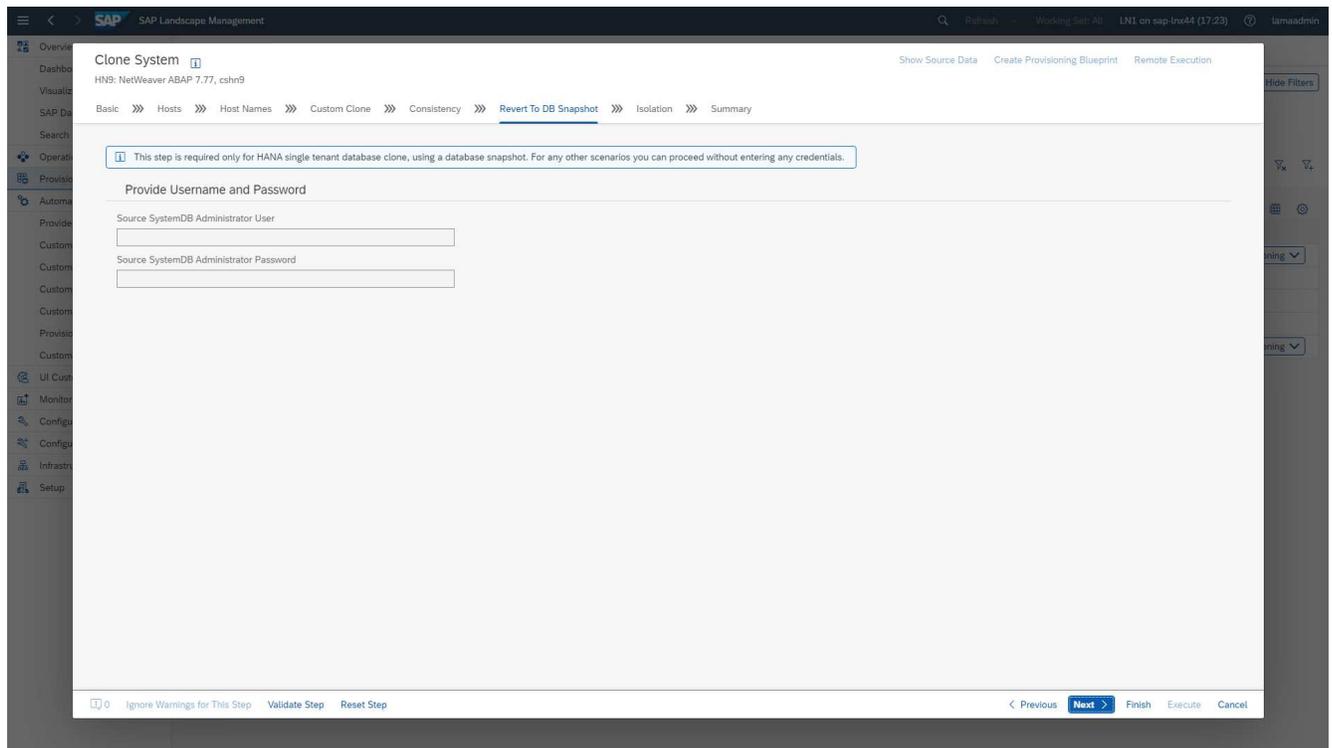
5. 在屏幕4中、将列出自定义克隆操作。提供了克隆和\* SnapPostfix\*名称、这些名称将在FlexClone卷和Snapshot名称的存储克隆操作期间使用。如果将这些字段留空、则为在提供程序脚本的变量部分中配置的默认值 netapp\_clone.sh 已使用。



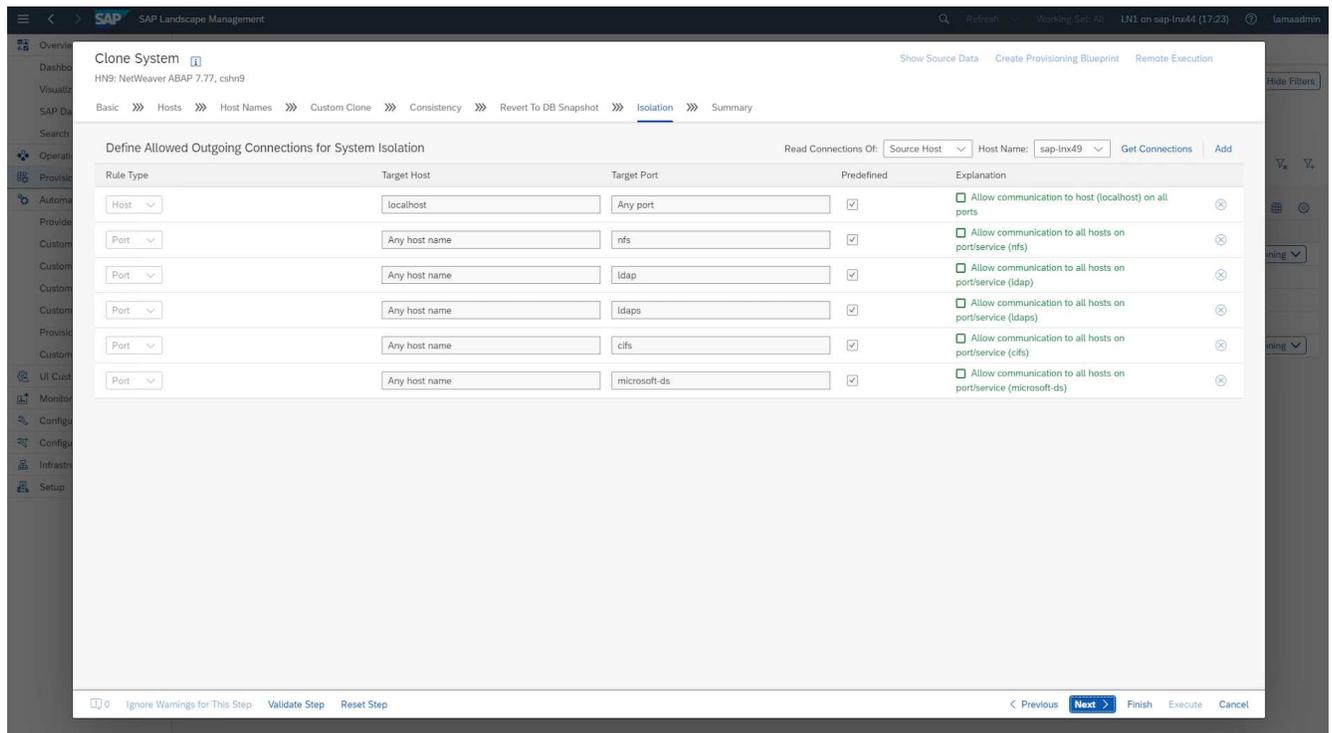
6. 在屏幕5中、已选择数据库一致性选项。在本示例中、我们选择了\*联机：克隆运行DB\*。



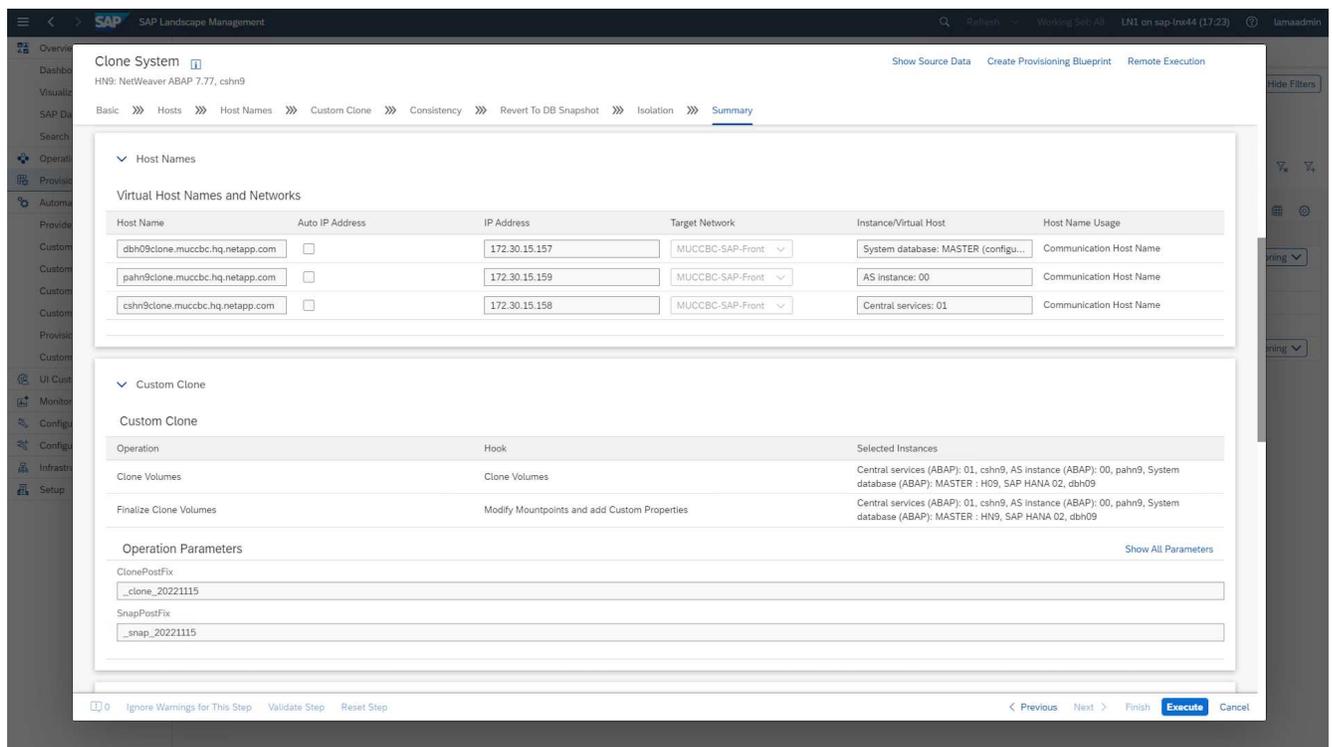
7. 在屏幕6中、只有在执行租户克隆时才需要输入。



8. 在屏幕7中、可以配置系统隔离。

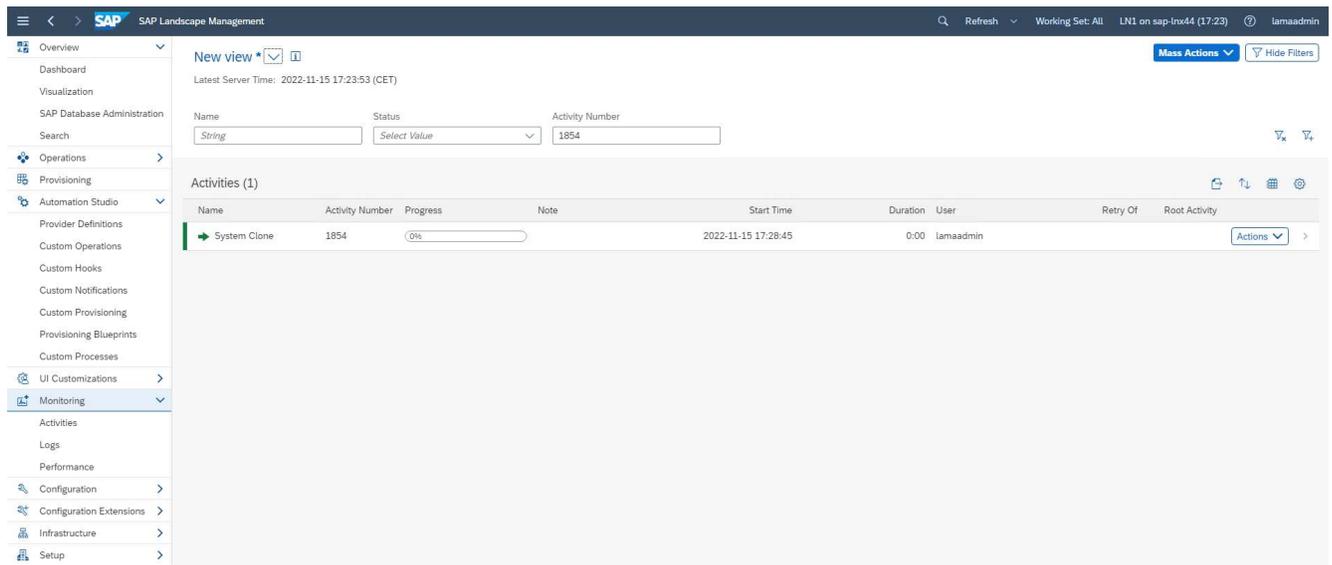


9. 在屏幕8中、摘要页面包含启动工作流之前进行最终确认的所有设置。单击\*执行\*以启动工作流。

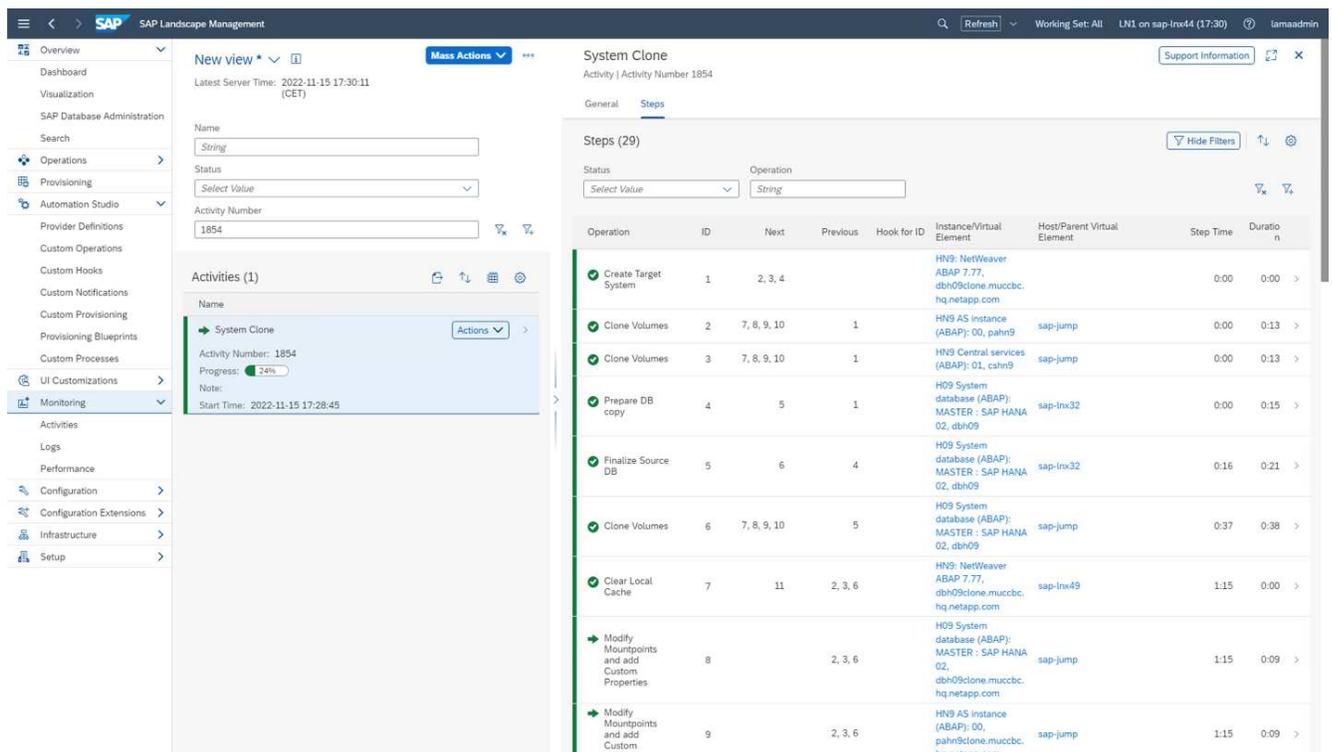


现在、SAP Lama将执行配置中指示的所有操作。这些操作包括创建存储卷克隆和导出、将其挂载到目标主机、添加用于隔离的防火墙规则以及启动HANA数据库和SAP服务。

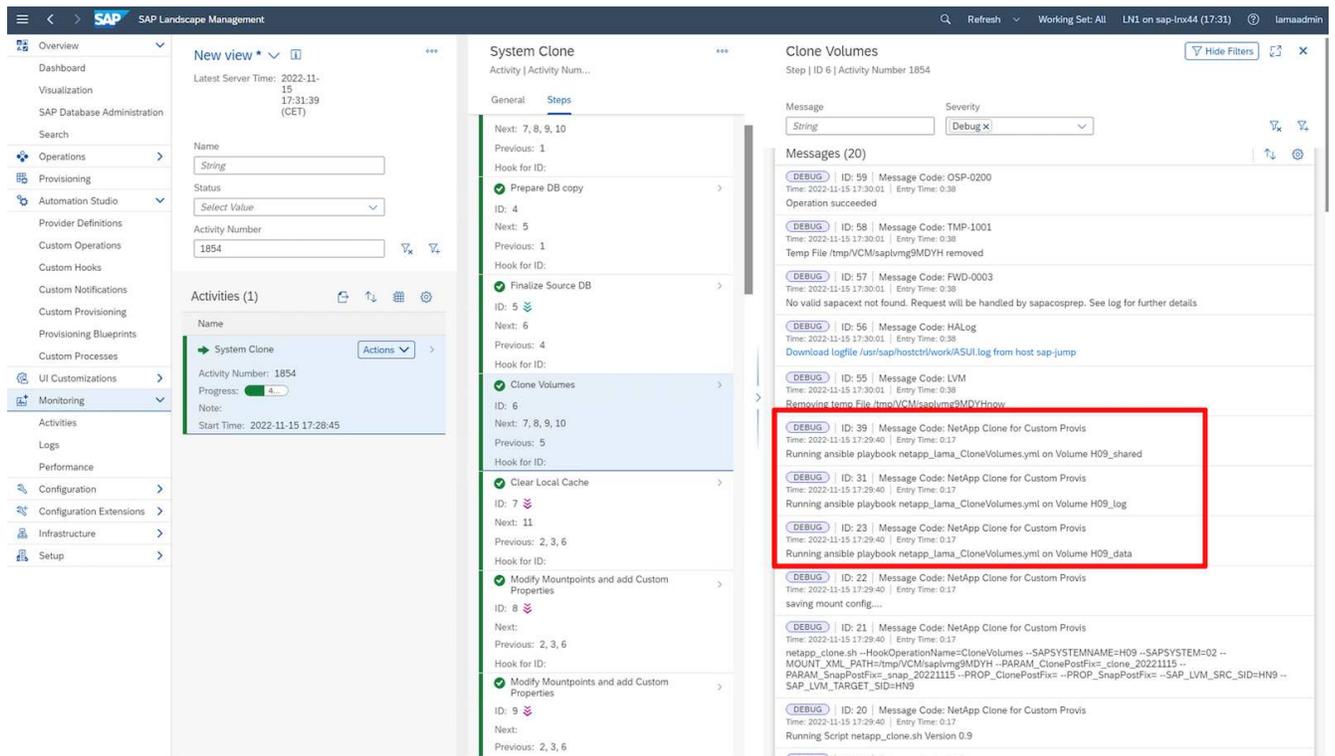
10. 您可以在\*监控\*菜单下监控克隆工作流的进度。



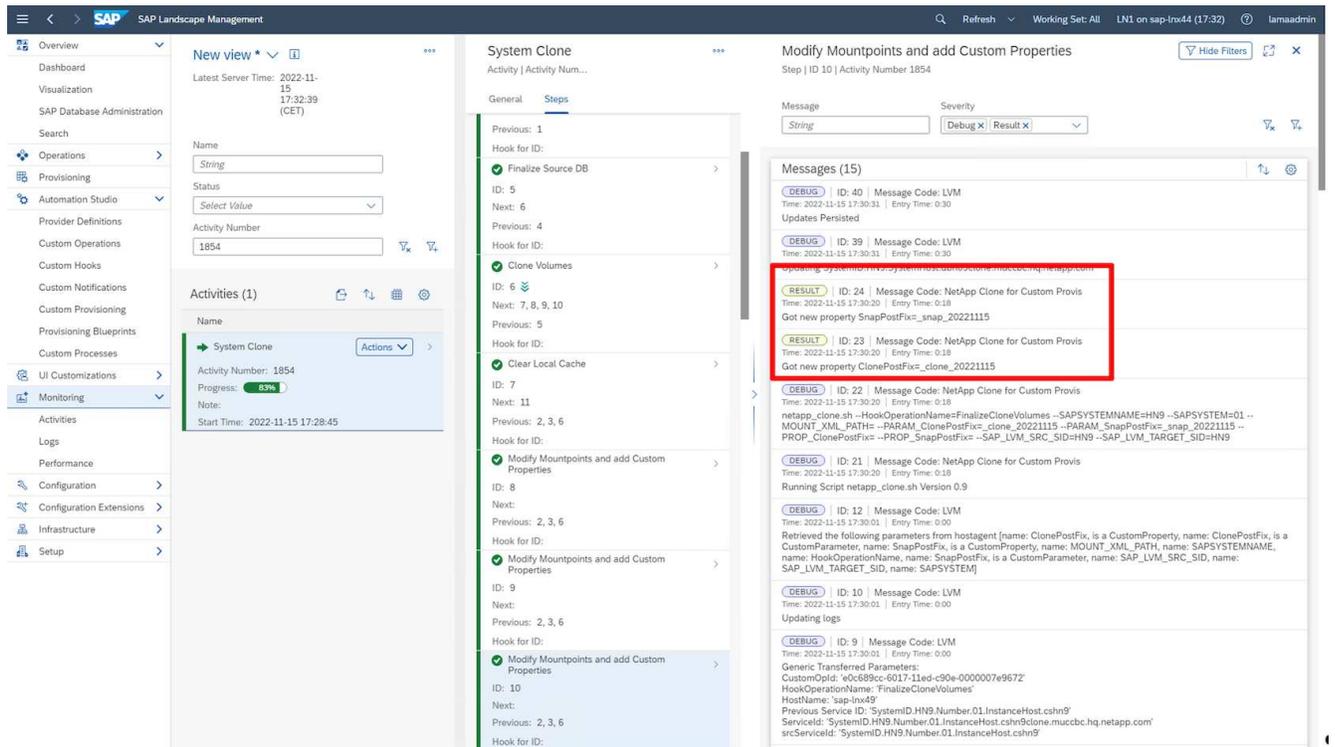
在详细日志中、操作\*克隆卷\*和\*修改挂载点并添加自定义属性\*将在Ansible节点上执行 sap-jump 主机。这些步骤会对每个服务、HANA数据库、SAP中央服务和SAP即服务执行。



11. 通过选择\*克隆卷\*任务、将显示该步骤的详细日志、并显示Ansible攻略手册的执行情况。您可以看到、Ansible攻略手册 netapp\_lama\_CloneVolumes.yml 对每个HANA数据库卷、数据、日志和共享执行。



12. 在\*修改挂载点并添加自定义属性\*步骤的详细信息视图中、您可以找到有关挂载点以及执行脚本移交的自定义属性的信息。



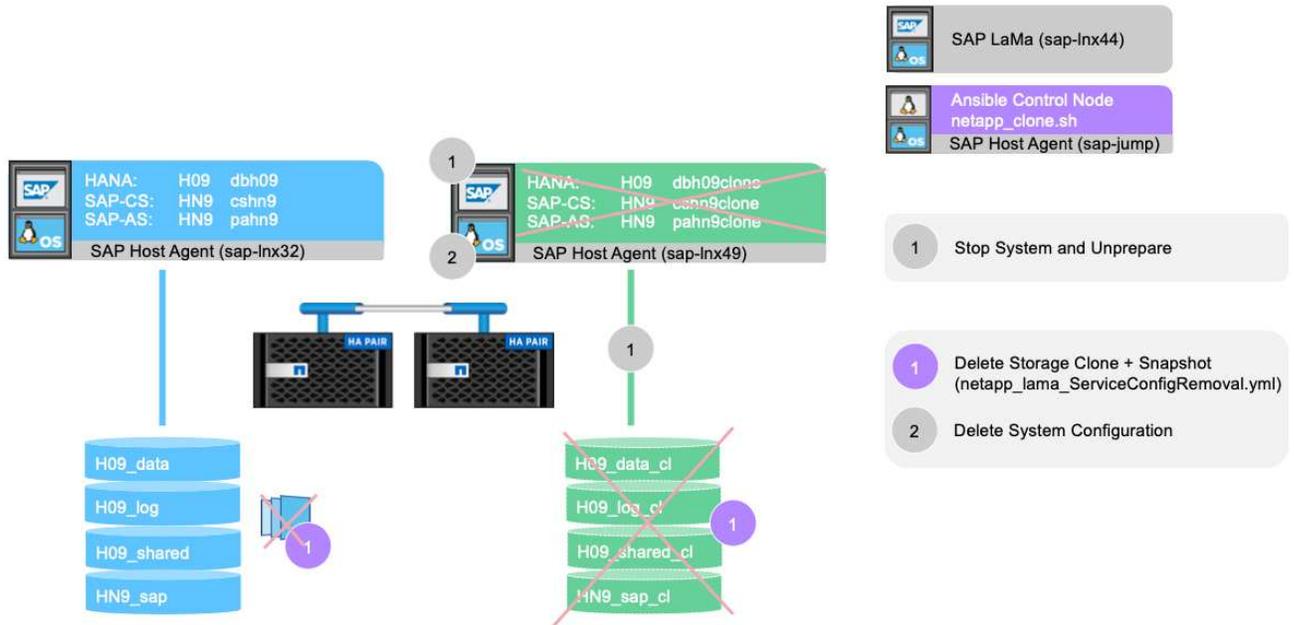
workflow完成后、克隆的SAP系统即已准备就绪、已启动并可供使用。

# SAP Lama取消配置 workflow—系统销毁

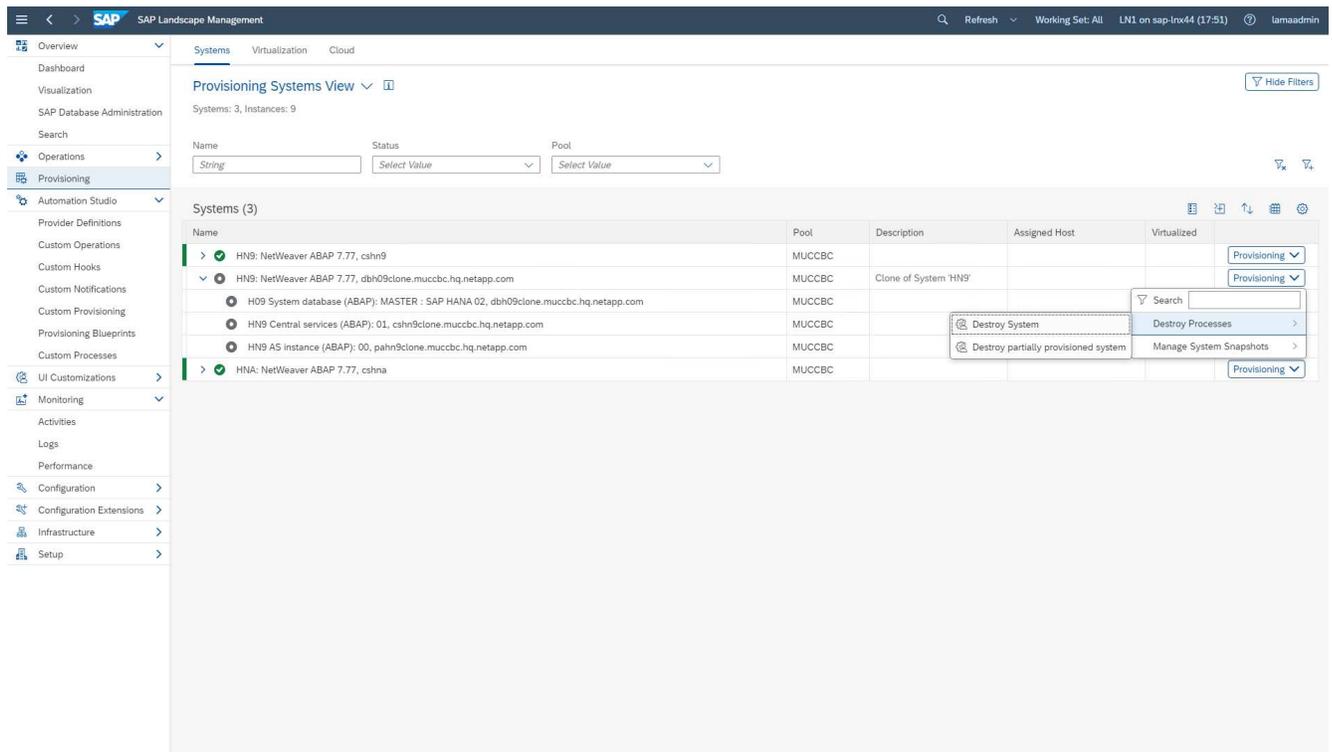
下图突出显示了使用系统销毁 workflow 执行的主要步骤。



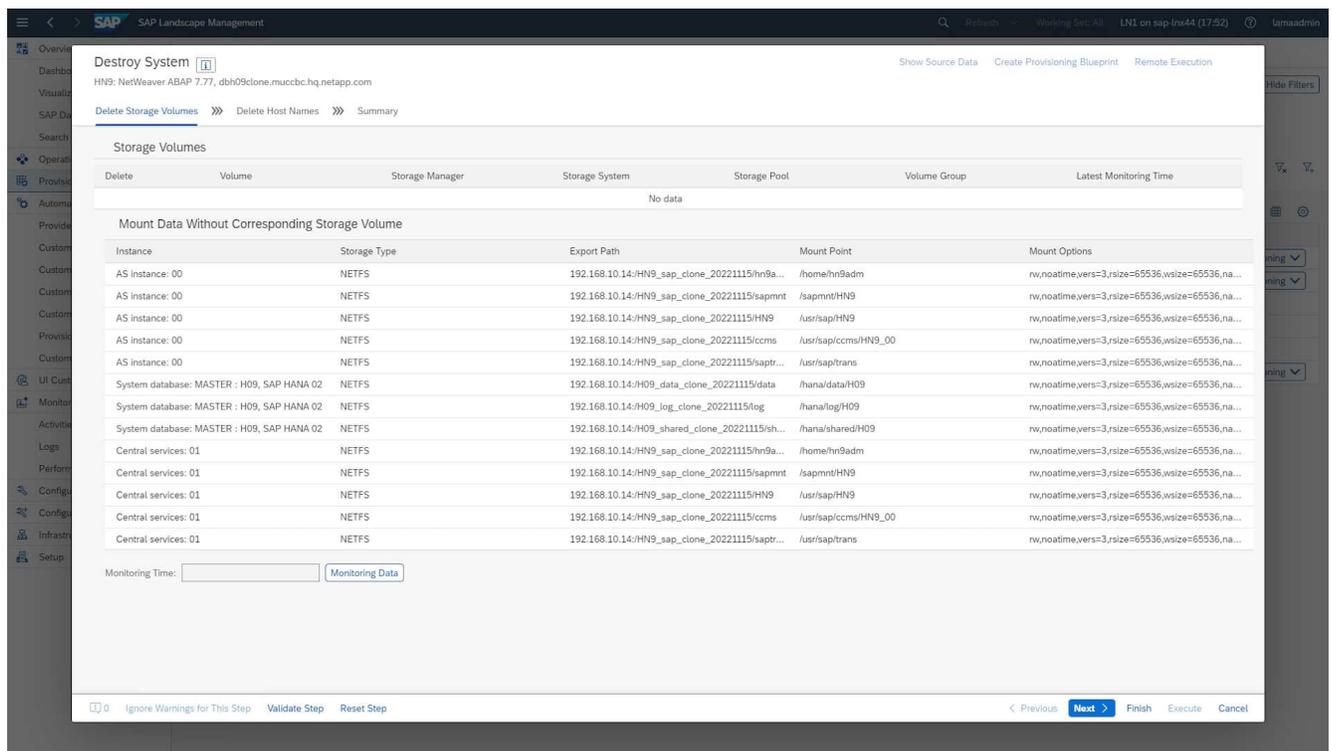
1. 要停用克隆的系统、必须事先停止并准备好该系统。之后、可以启动系统销毁 workflow。



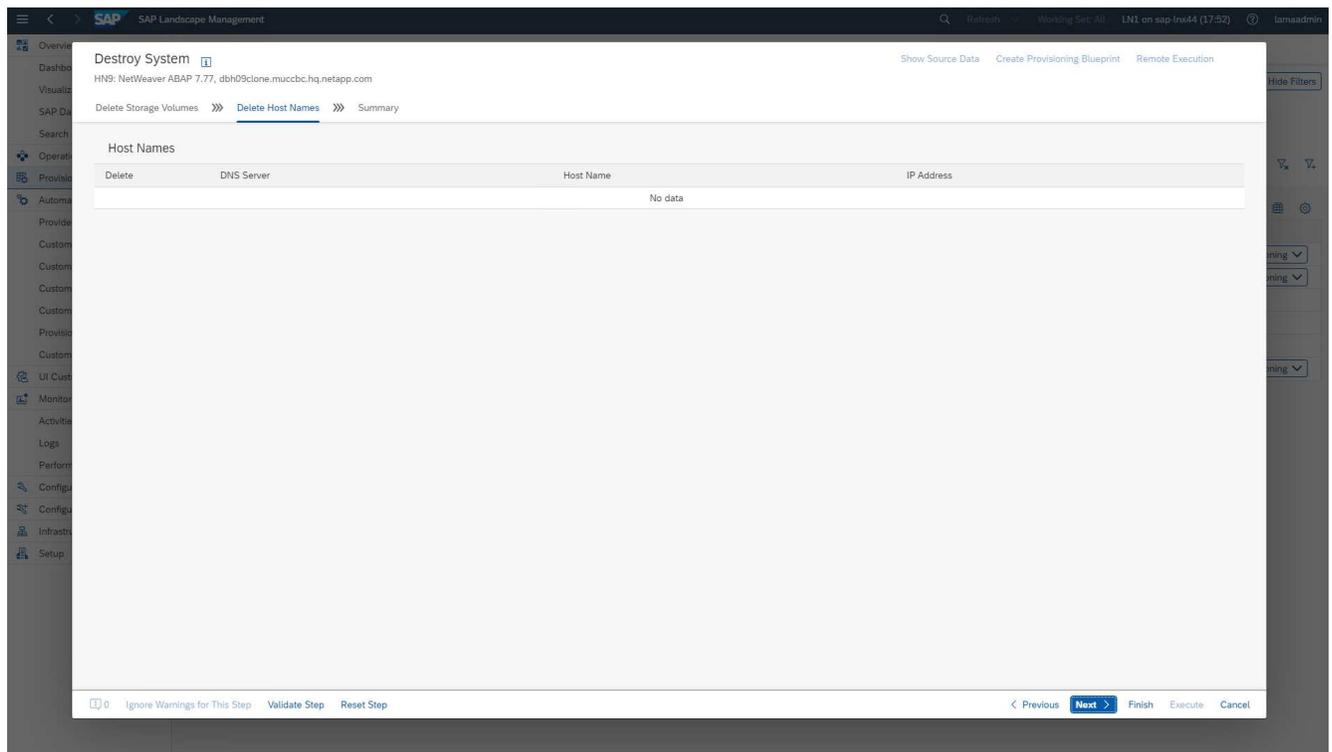
2. 在此示例中、我们将为之前创建的系统运行系统销毁 workflow。我们在\*系统视图\*屏幕中选择系统、然后在\*销毁进程\*下启动系统销毁 workflow。



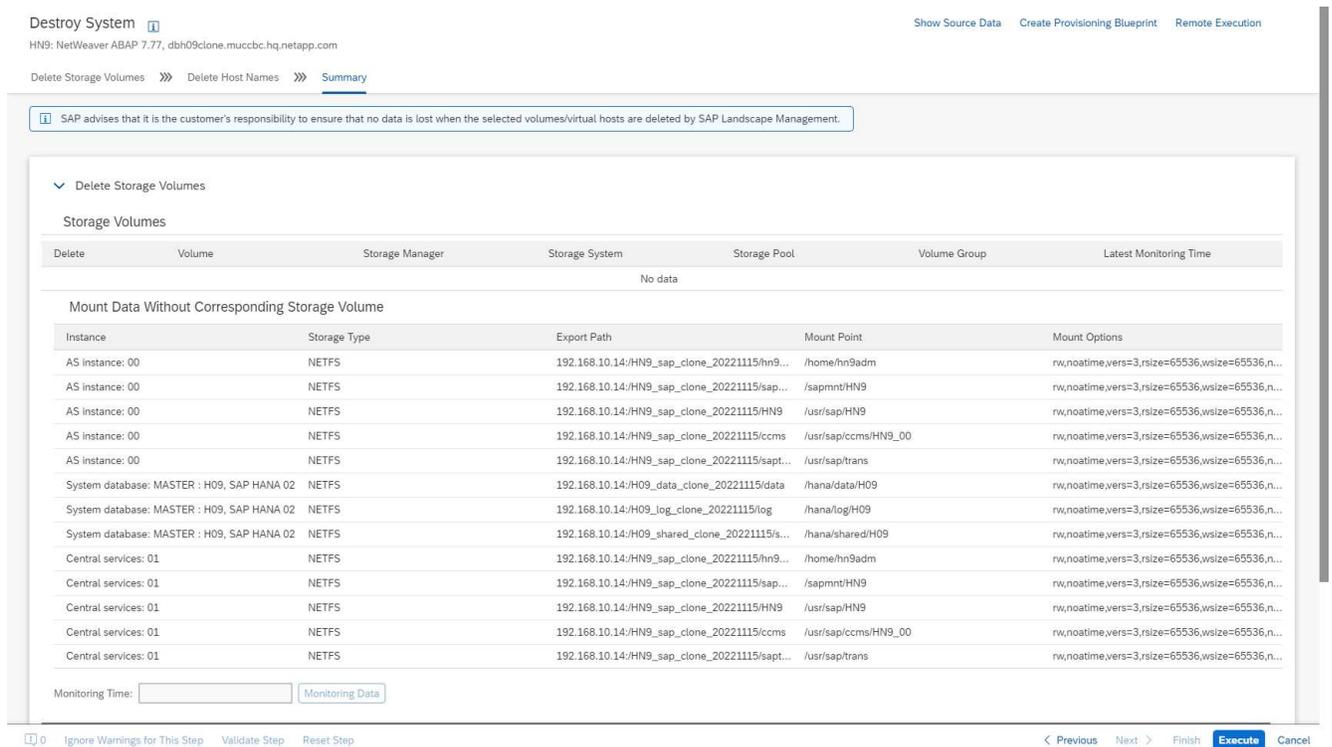
3. 此处显示了配置阶段维护的所有挂载点、并在系统销毁工作流程过程中删除这些挂载点。



不会删除任何虚拟主机名、因为这些主机名通过DNS维护并自动分配。

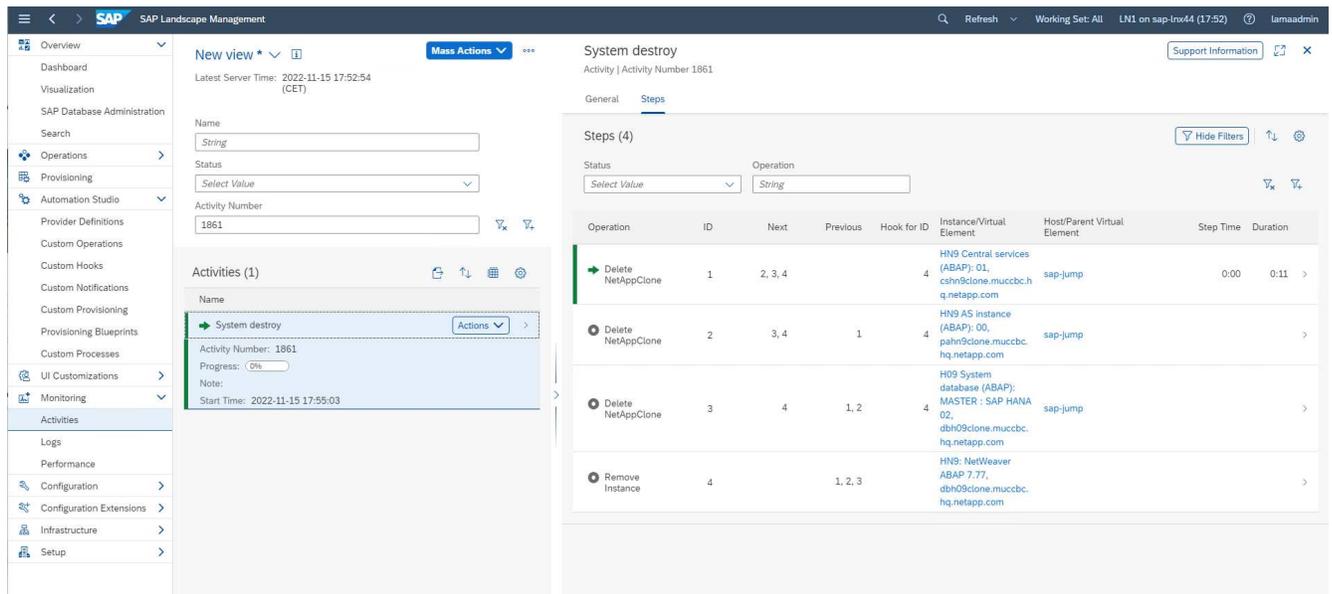


4. 单击执行按钮可启动此操作。

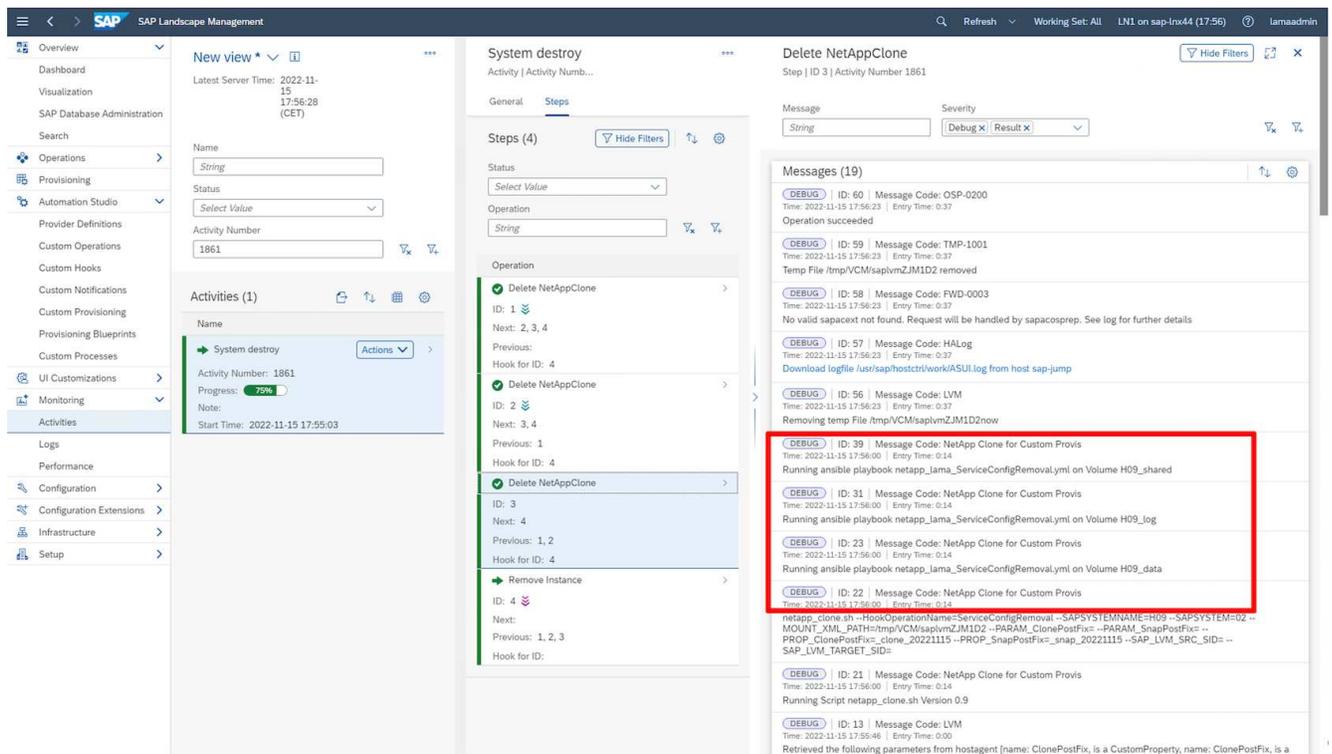


现在、SAP Lama会删除卷克隆并删除克隆系统的配置。

5. 您可以在\*监控\*菜单下监控克隆工作流的进度。

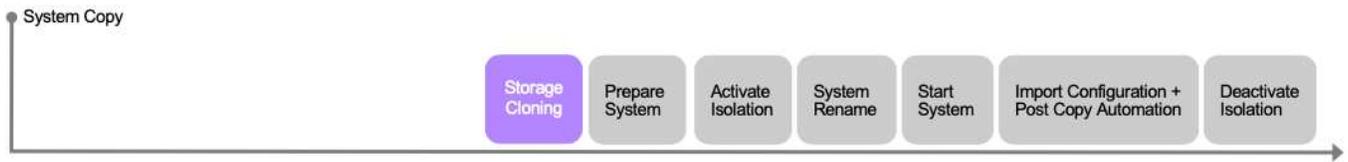


6. 通过选择\*删除NetAppClone\*任务、将显示该步骤的详细日志。此处显示了Ansible攻略手册的执行情况。如您所见、《Ansible攻略手册》netapp\_lama\_ServiceConfigRemoval.yml 对每个HANA数据库卷、数据、日志和共享执行。

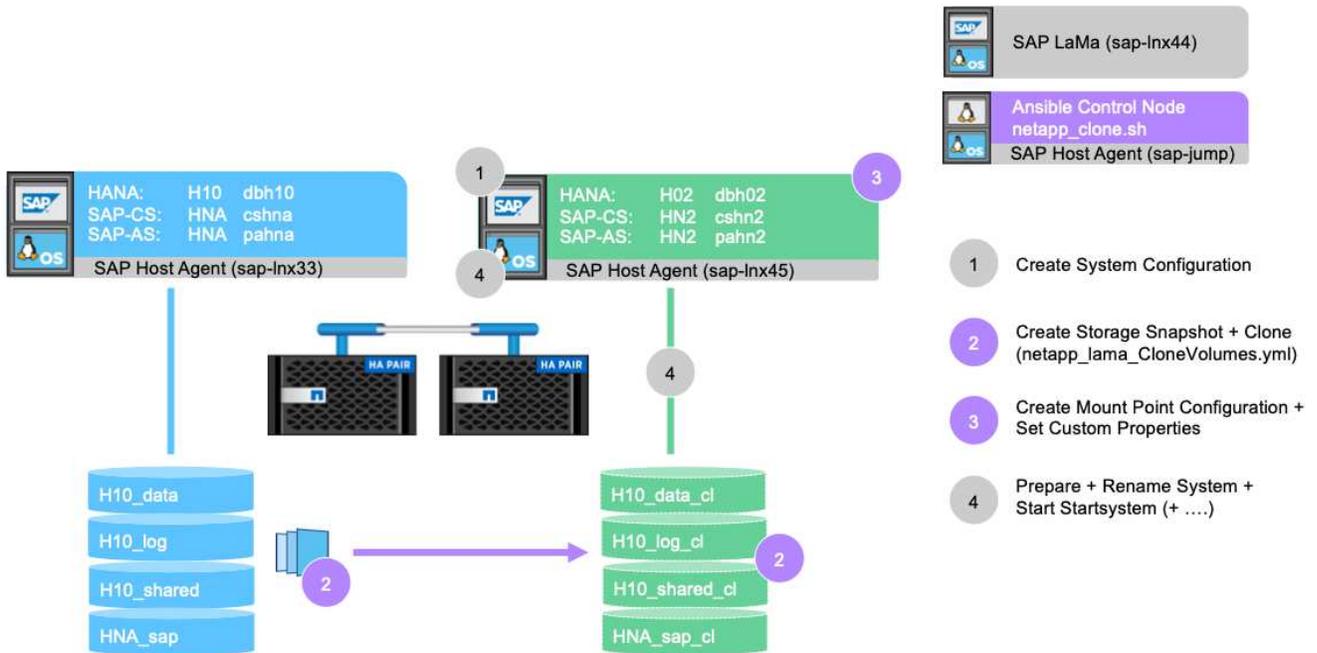


## SAP Lama配置 workflow—复制系统

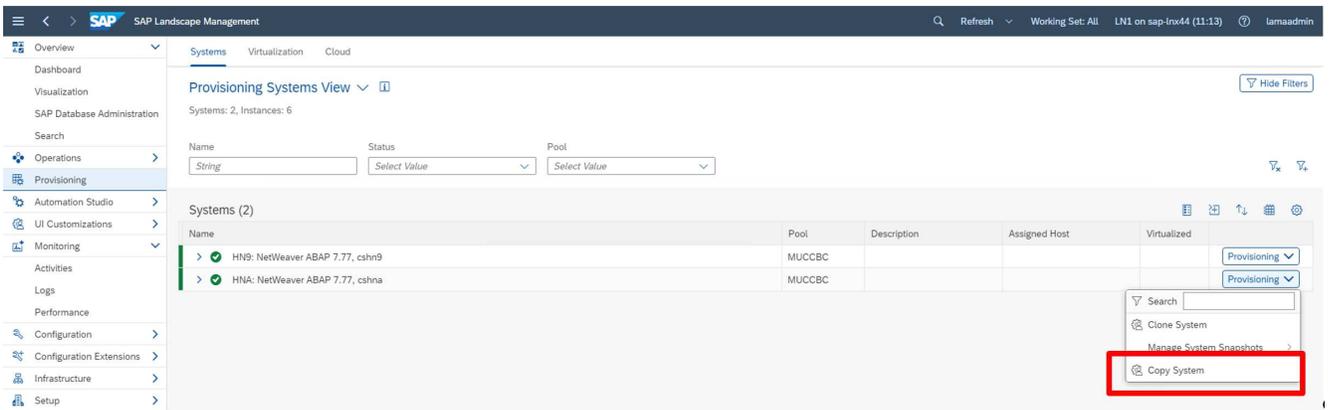
下图突出显示了使用系统副本 workflow 执行的主要步骤。



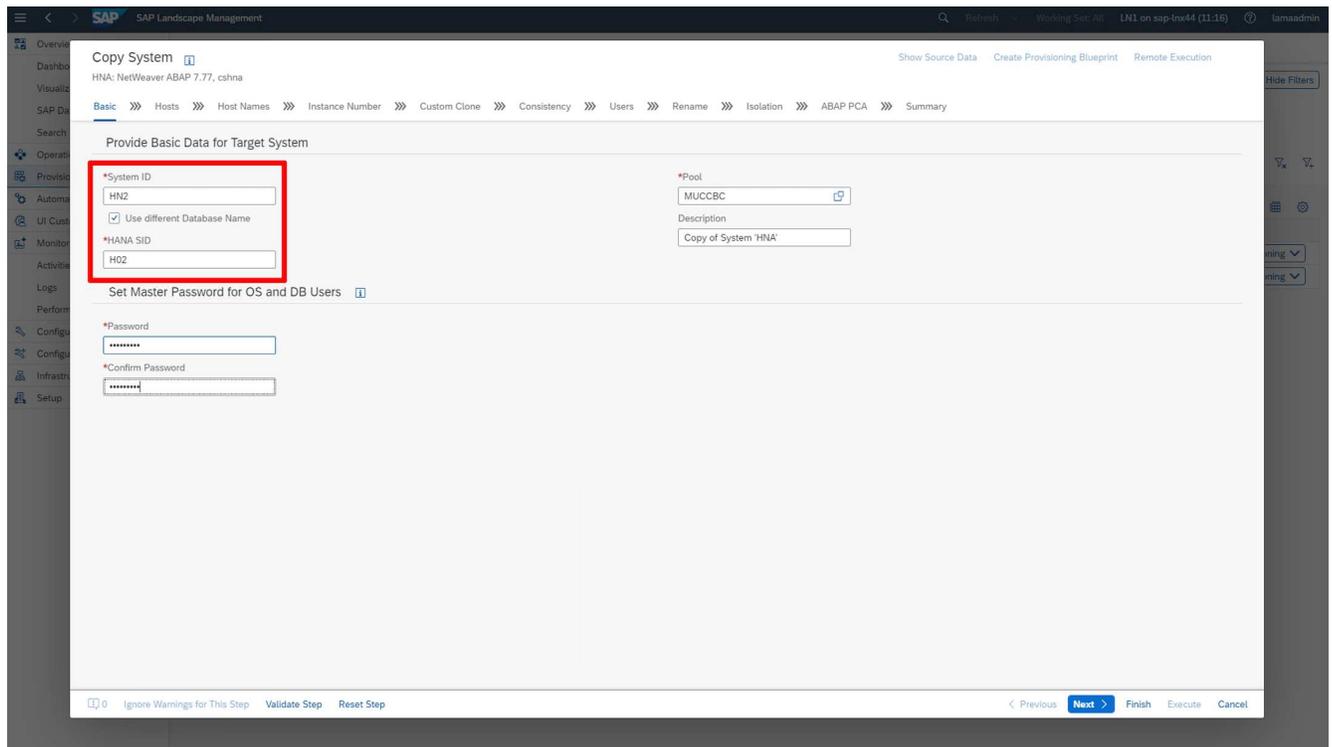
在本章中、我们将简要讨论系统克隆 workflow 和输入屏幕的不同之处。如下图所示、存储 workflow 不会发生任何变化。



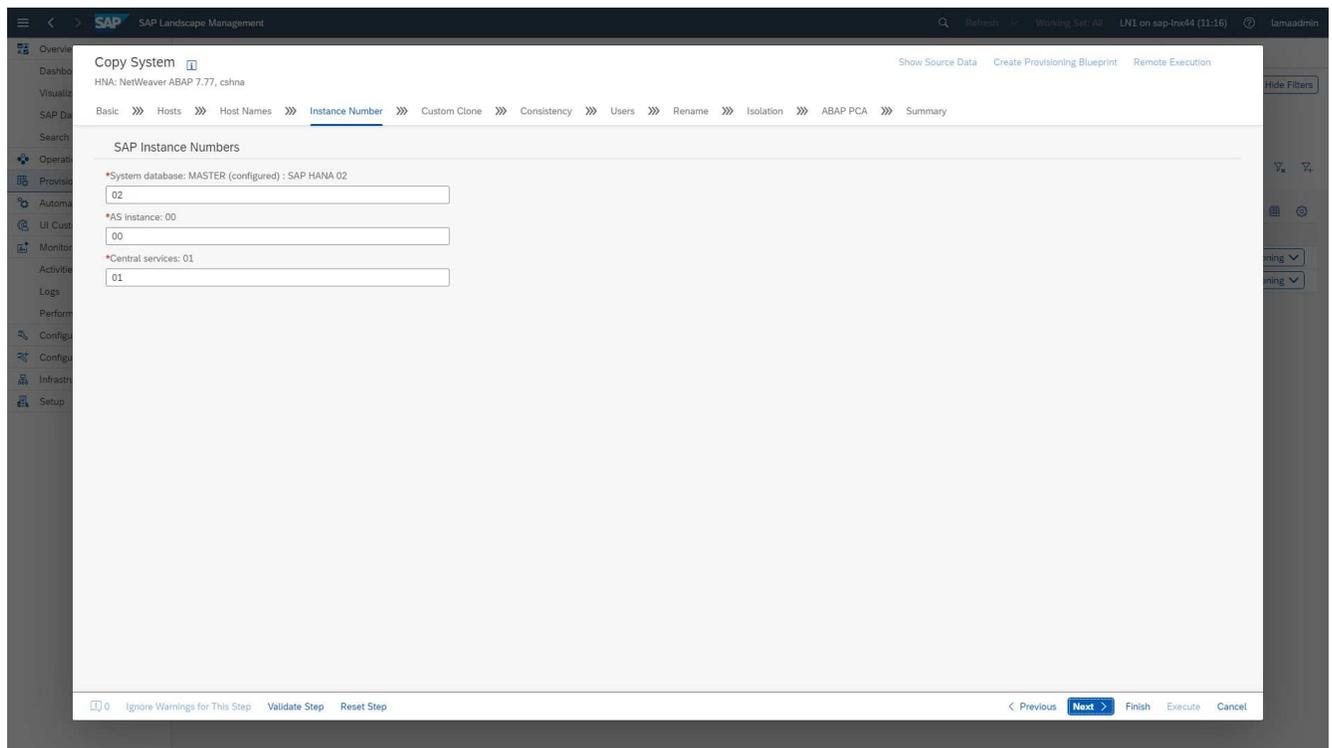
1. 系统副本 workflow 可以在系统准备就绪后启动。这不是此配置的特定任务、我们也不会对此进行详细说明。如果您需要更多信息、请查看 SAP Lama 文档。



2. 在复制 workflow 期间、系统将根据第一个屏幕中的指定进行重命名。

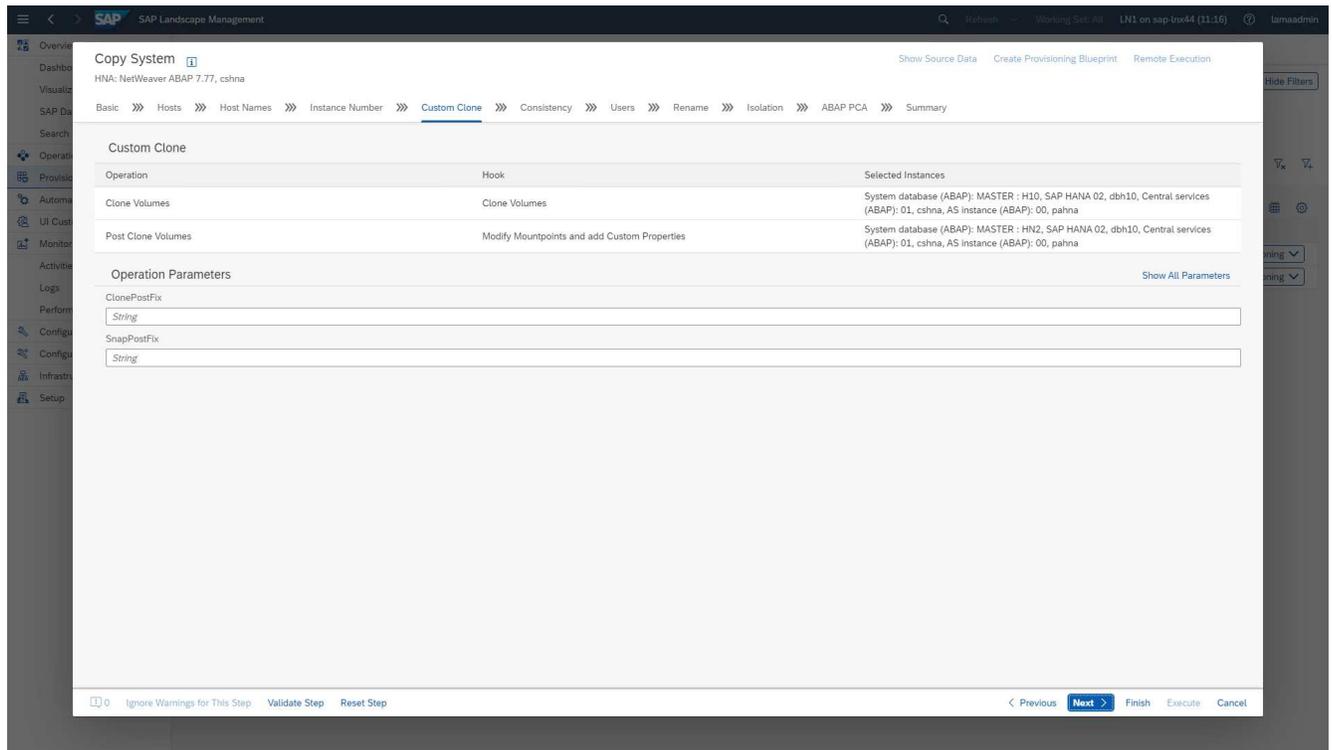


3. 在工作流期间、您可以更改实例编号。

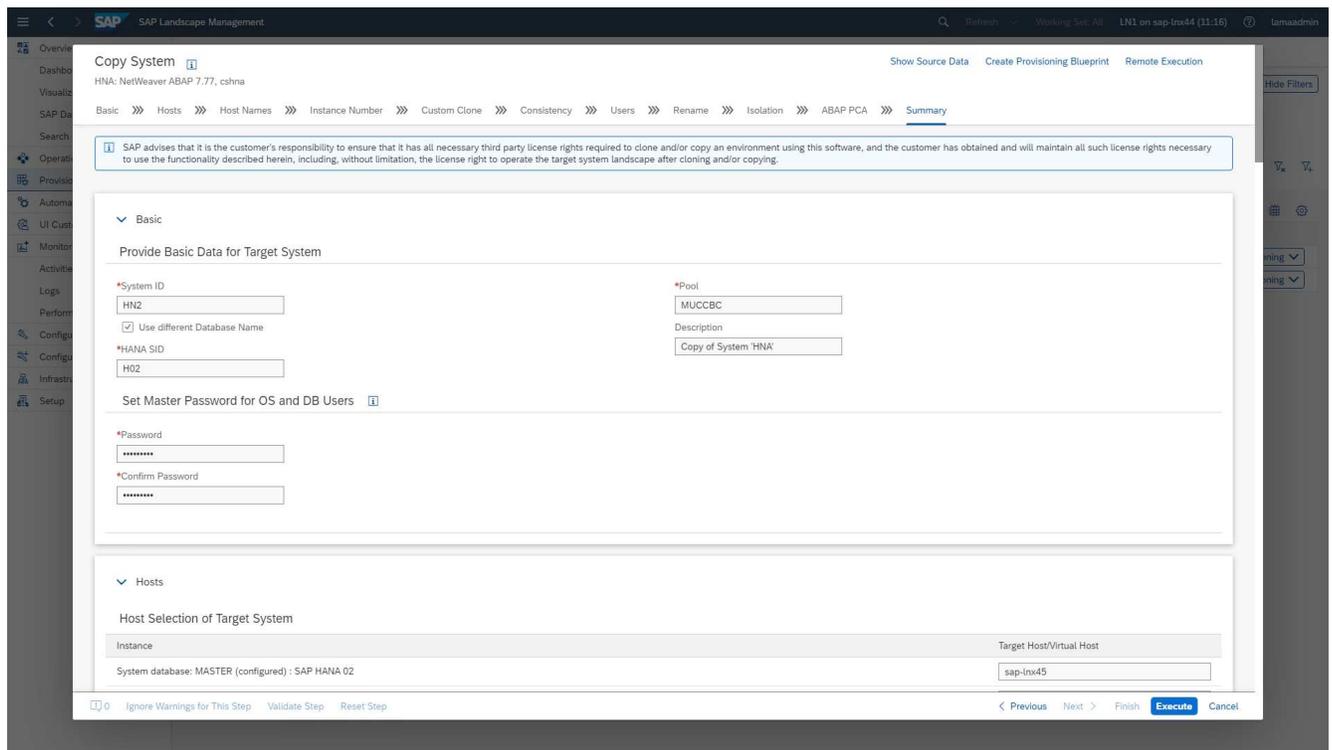


更改实例编号尚未经过测试、可能需要在提供程序脚本中进行更改。

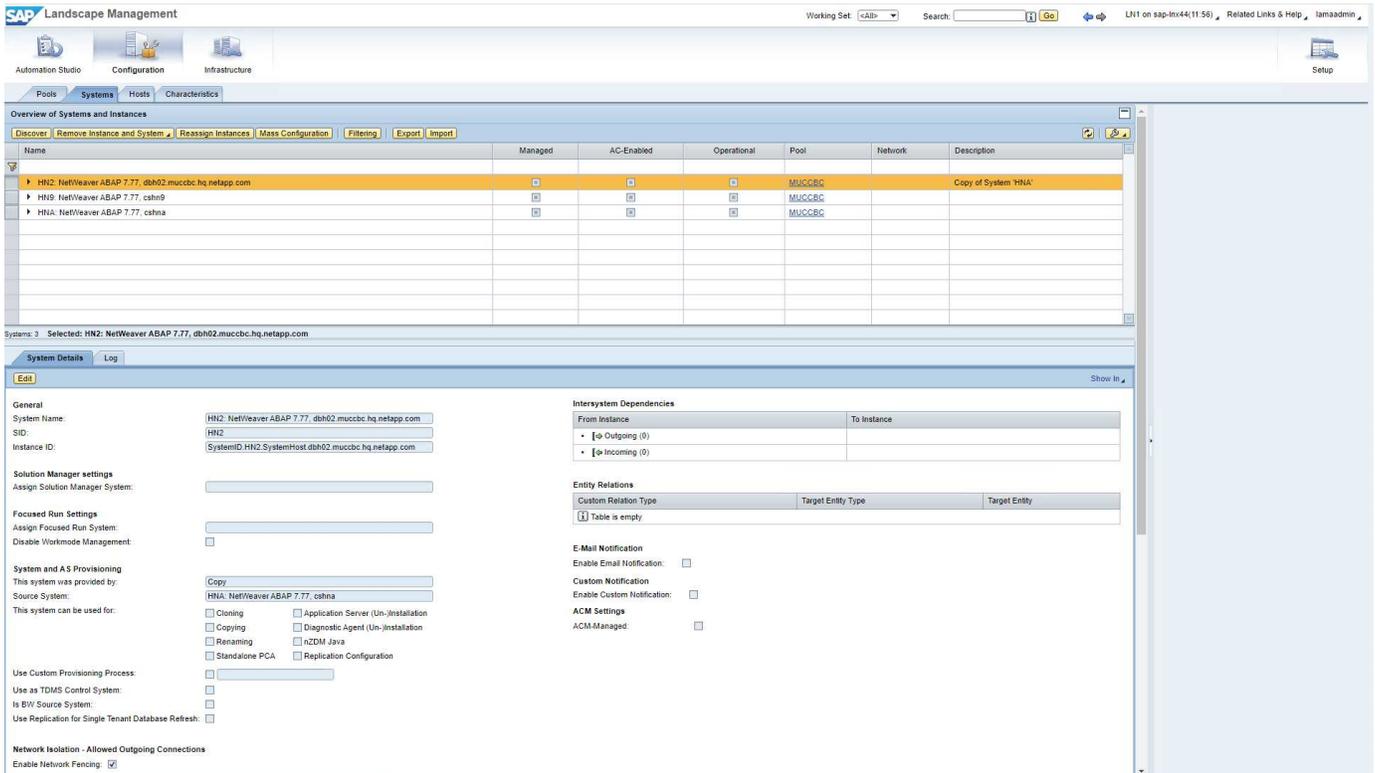
4. 如上所述、\*自定义克隆\*屏幕与克隆工作流并无不同、如下所示。



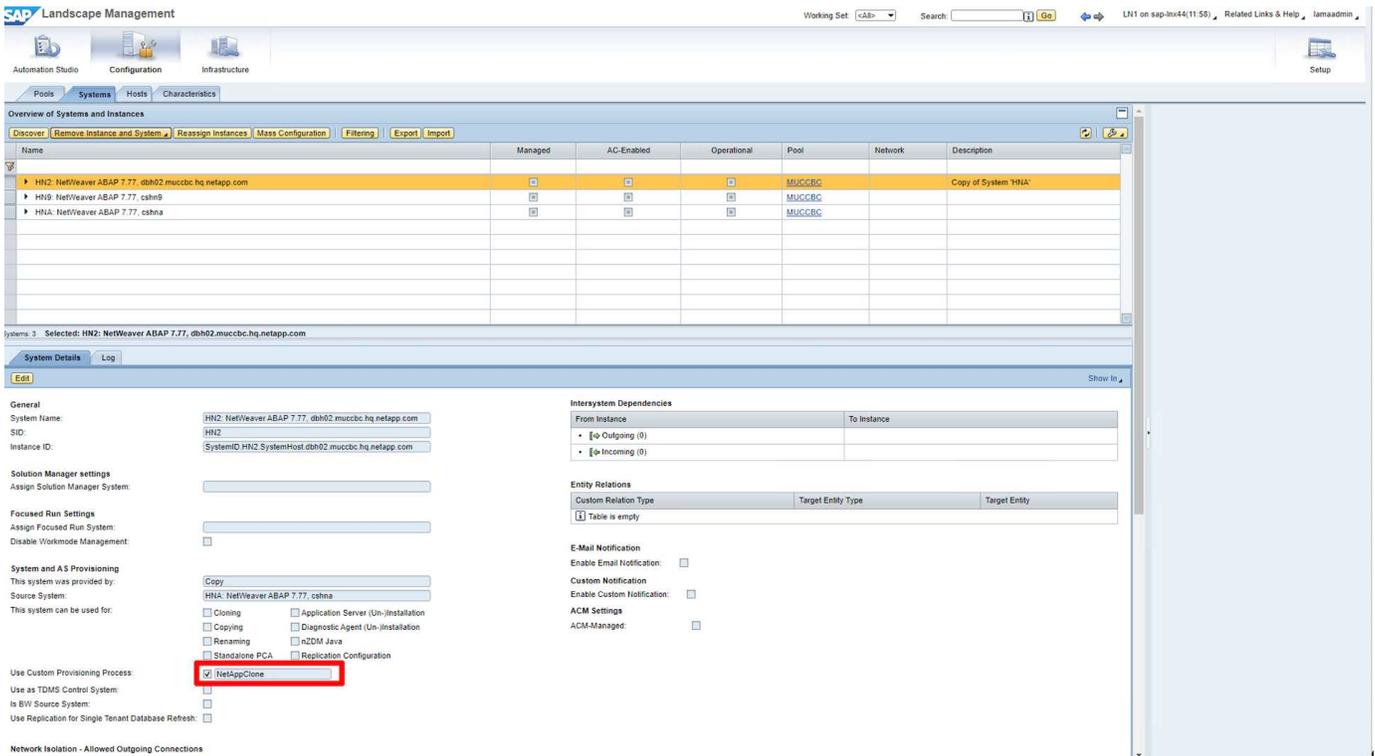
5. 如我们所述、其余输入掩码不会与标准有所偏差、我们不会在此处进一步介绍这些掩码。最后一个屏幕将显示摘要、现在可以开始执行。



复制过程完成后、不会为自定义克隆过程启用目标实例。

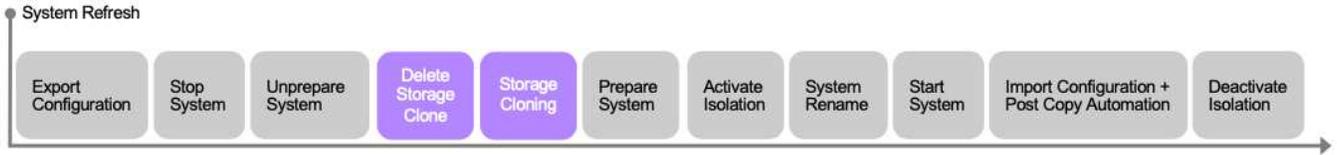


在系统销毁过程中、必须手动采用预挂步骤、因为已设置限制并会阻止执行。

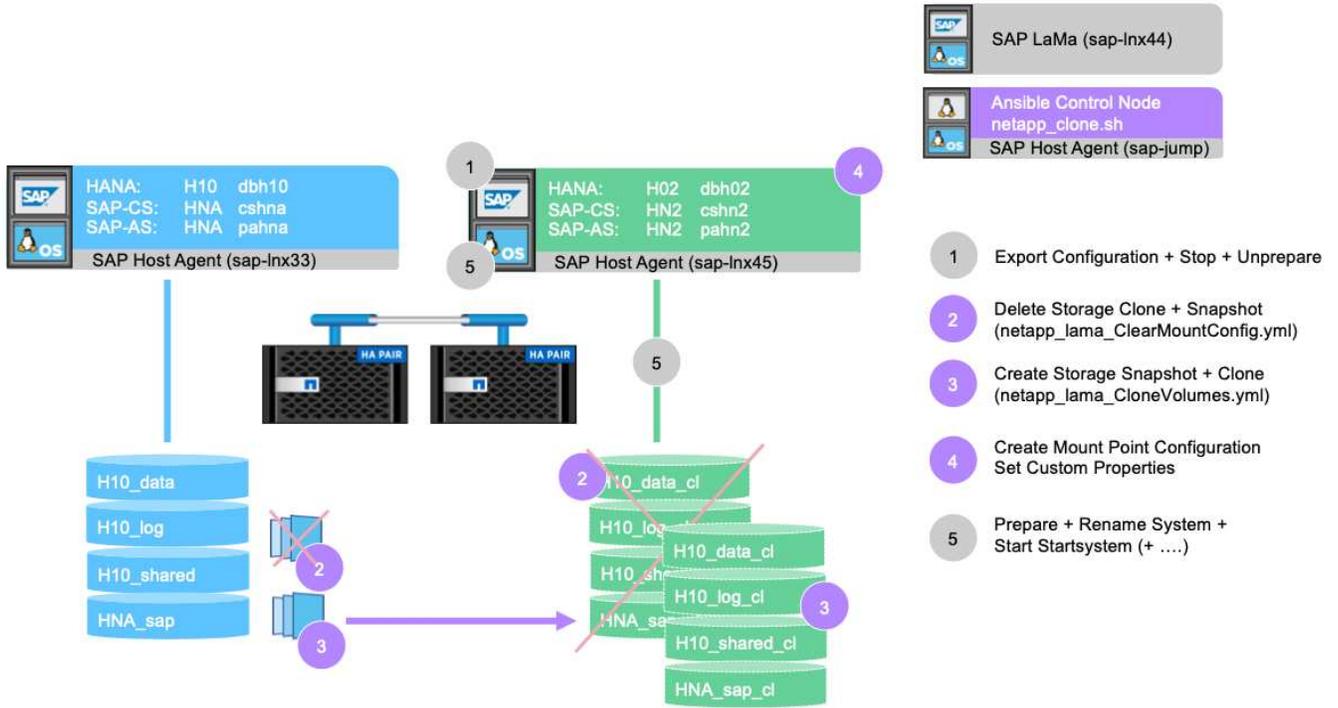


## SAP Lama配置 workflow—系统刷新

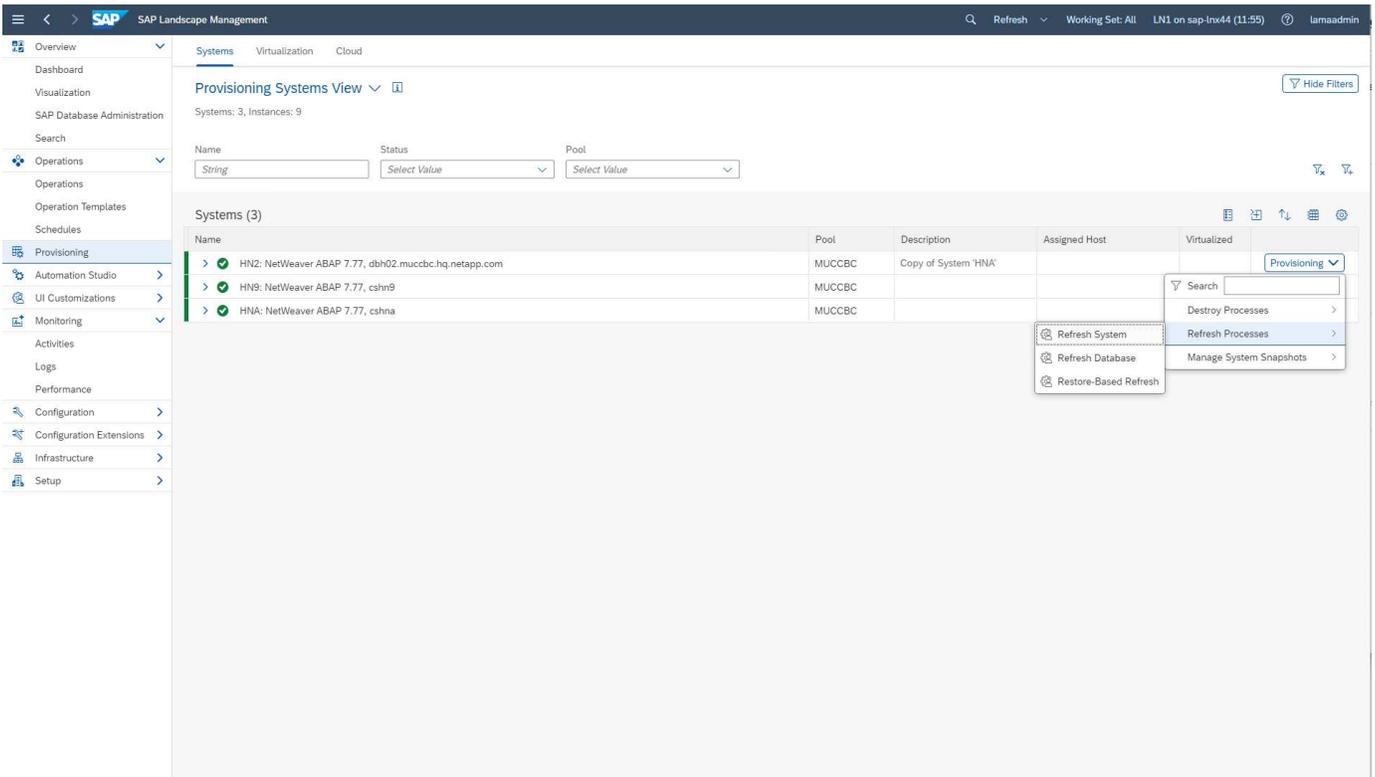
下图突出显示了使用系统刷新 workflow 执行的主要步骤。



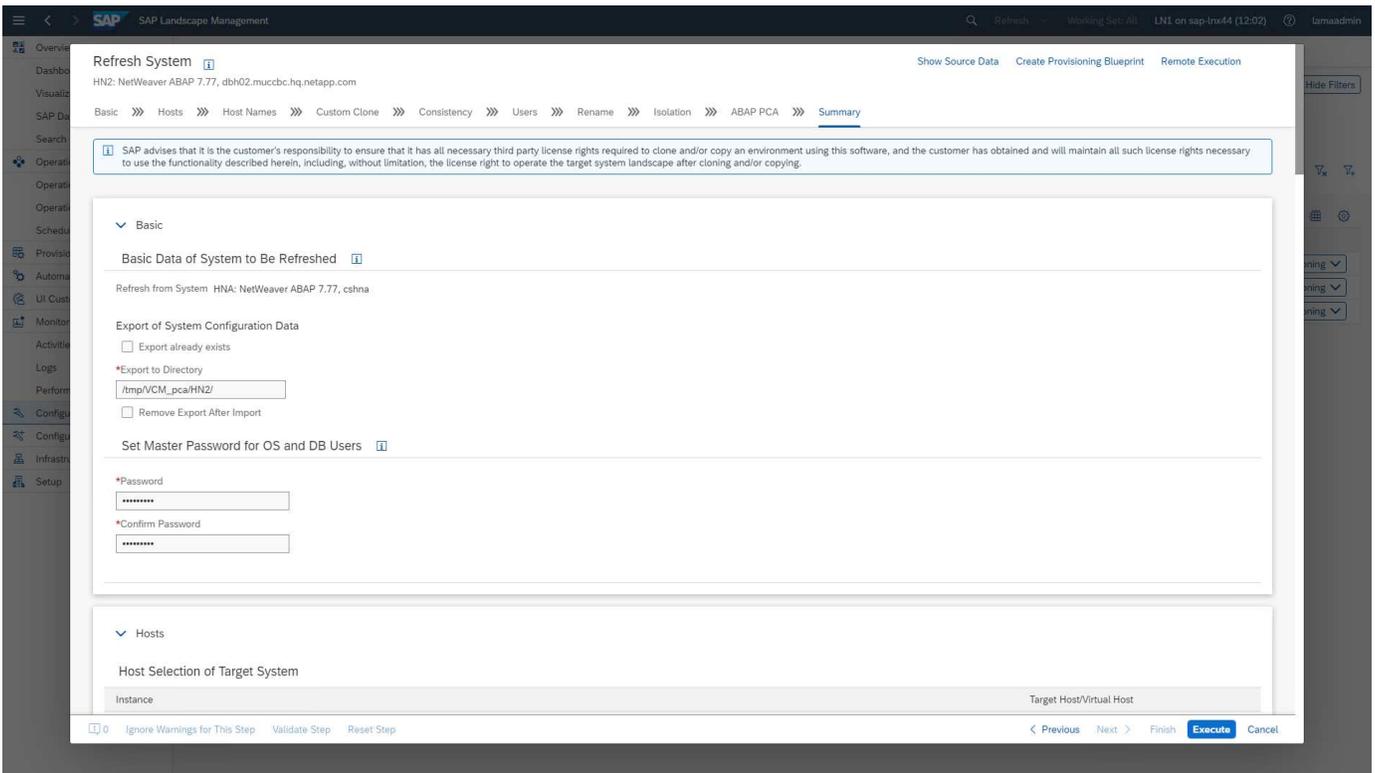
在刷新工作流期间、必须删除存储克隆。您可以使用与系统销毁工作流相同的Ansible攻略手册。但是、自定义挂锁的定义步骤与此不同、因此攻略手册的名称也相应。克隆的n't步骤与此不同。



刷新工作流可以通过复制的系统的配置屏幕触发。



同样、输入屏幕与标准屏幕没有任何不同、可以从摘要屏幕开始执行工作流。



## 提供程序脚本配置和Ansible攻略手册

在本文档中的示例部署和工作流执行期间、我们将使用以下提供程序配置文件、执行脚本

和Ansible攻略手册。



示例脚本按原样提供、NetApp不支持。您可以通过电子邮件发送电子邮件至：[mailto: ng-sapcc@netapp.com](mailto:ng-sapcc@netapp.com) [ng-sapcc@netapp.com](mailto:ng-sapcc@netapp.com)^请求最新版本的脚本。

## 提供程序配置文件netapp\_clone.conf

此时将按照中所述创建配置文件 "[SAP Lama文档—配置SAP Host Agent注册脚本](#)"。此配置文件必须位于安装了SAP主机代理的Ansible控制节点上。

已配置的操作系统用户 `sapuser` 必须具有执行脚本和所调用的Ansible攻略手册的适当权限。您可以将脚本放置在通用脚本目录中。SAP Lama可以在调用脚本时提供多个参数。

除了自定义参数之外、`PARAM_ClonePostFix`、`PROP_ClonePostFix`、`PARAM_ClonePostFix`、和 `PROP_ClonePostFix`、可以移交许多其他人、如中所示 "[SAP Lama文档](#)"。

```
root@sap-jump:~# cat /usr/sap/hostctrl/exe/operations.d/netapp_clone.conf
Name: netapp_clone
Username: sapuser
Description: NetApp Clone for Custom Provisioning
Command: /usr/sap/scripts/netapp_clone.sh
--HookOperationName=${HookOperationName} --SAPSYSTEMNAME=${SAPSYSTEMNAME}
--SAPSYSTEM=${SAPSYSTEM} --MOUNT_XML_PATH=${MOUNT_XML_PATH}
--PARAM_ClonePostFix=${PARAM_ClonePostFix} --PARAM_SnapPostFix=${PARAM
-SnapPostFix} --PROP_ClonePostFix=${PROP_ClonePostFix}
--PROP_SnapPostFix=${PROP_SnapPostFix}
--SAP_LVM_SRC_SID=${SAP_LVM_SRC_SID}
--SAP_LVM_TARGET_SID=${SAP_LVM_TARGET_SID}
ResulConverter: hook
Platform: Unix
```

## 提供程序脚本netapp\_clone.sh

提供程序脚本必须存储在中 `/usr/sap/scripts` 在提供程序配置文件中配置。

### 变量

以下变量在脚本中进行了硬编码、必须进行相应调整。

- `PRIMARY_CLUSTER=<hostname of netapp cluster>`
- `PRIMARY_SVM=<SVM name where source system volumes are stored>`

证书文件 `PRIMARY_KEYFILE=/usr/sap/scripts/ansible/certs/ontap.key` 和 `PRIMARY_CERTFILE=/usr/sap/scripts/ansible/certs/ontap.pem` 必须按照中所述提供 "[NetApp Ansible模块—准备ONTAP](#)"。



如果不同的SAP系统需要不同的集群或SVM、则可以在SAP Lama提供程序定义中将这此变量作为参数添加。

功能：创建清单文件

要使Ansible攻略手册的执行更具动态性、请使用 `inventory.yml` 文件会即时创建。在变量部分中配置了一些静态值、而在执行期间动态创建了一些静态值。

功能：运行Ansible攻略手册

此功能用于将Ansible攻略手册与动态创建的一起执行 `inventory.yml` 文件攻略手册的命名约定为 `netapp_lama_{{HookOperationName}}.yml`。的值 `{{HookOperationName}}` 依赖于Lama操作、并由Lama作为命令行参数进行移交。

主部分

本节包含主要执行计划。变量 `{{HookOperationName}}` 包含Lama替换步骤的名称、由Lama在调用该脚本时提供。

- 系统克隆和系统副本配置工作流的价值：
  - 克隆卷
  - `PostCloneVolumes`
- 系统销毁工作流的价值：
  - `ServiceConfigRemoval`
- 系统刷新工作流的价值：
  - `ClearMountConfig`

**HookOperationName = 克隆卷**

执行此步骤后、将执行Ansible攻略手册、从而触发Snapshot副本和克隆操作。卷名称和挂载配置由SAP Lama通过变量中定义的XML文件传递 `$MOUNT_XML_PATH`。此文件已保存、因为此文件将在步骤的稍后部分使用 `FinalizeCloneVolumes` 以创建新的挂载点配置。卷名称将从XML文件中提取、并对每个卷执行Ansible克隆攻略手册。



在此示例中、作为实例和中央服务共享同一个卷。因此、只有在使用SAP实例编号时才会执行卷克隆 (`$SAPSYSTEM`不是 01)。在其他环境中、这可能有所不同、必须相应地进行更改。

**HookOperationName = PostCloneVolumes**

在此步骤中、将显示自定义属性 `ClonePostFix` 和 `SnapPostFix` 并维护目标系统的挂载点配置。

在期间停用系统时、自定义属性稍后会用作输入 `ServiceConfigRemoval` 或 `ClearMountConfig` 阶段。系统用于保留在系统配置工作流期间指定的自定义参数设置。

此示例中使用的值为 `ClonePostFix=_clone_20221115` 和 `SnapPostFix=_snap_20221115`。

卷 `HN9_sap`、动态创建的Ansible文件包括以下值：`datavolumename: HN9_sap`，`snapshotpostfix: _snap_20221115`，和 `clonepostfix: _clone_20221115`。

这将导致卷HN9\_SAP上的快照名称 HN9\_sap\_snap\_20221115 以及创建的卷克隆名称 HN9\_sap\_clone\_20221115。



可以使用任何自定义属性来保留配置过程中使用的参数。

挂载点配置将从中的Lama移交的XML文件中提取 CloneVolume 步骤。。 ClonePostFix 添加到卷名称中、并通过默认脚本输出发送回Lama。中介绍了此功能 "[SAP注释1889590](#)"。



在此示例中、存储系统上的qtree是在单个卷上放置不同数据的常用方法。例如：HN9\_sap 保存的挂载点 /usr/sap/HN9, /sapmnt/HN9, 和 /home/hn9adm。子目录的工作方式相同。在其他环境中、这可能有所不同、必须相应地进行更改。

### HookOperationName = ServiceConfigRemoval

在此步骤中、负责删除卷克隆的Ansible攻略手册正在运行。

卷名称由SAP Lama通过挂载配置文件和自定义属性进行传递 ClonePostFix 和 SnapPostFix 用于传递系统配置工作流程期间最初指定的参数值(请参见上的注释 HookOperationName = PostCloneVolumes) 。

卷名称将从xml文件中提取、并对每个卷执行Ansible克隆攻略手册。



在此示例中、作为实例和中央服务共享同一个卷。因此、只有在使用SAP实例编号时才会执行卷删除 (\$SAPSYSTEM)不是 01。在其他环境中、这可能有所不同、必须相应地进行更改。

### HookOperationName = ClearMountConfig

在此步骤中、负责在系统刷新工作流程期间删除卷克隆的Ansible攻略手册正在运行。

卷名称由SAP Lama通过挂载配置文件和自定义属性进行传递 ClonePostFix 和 SnapPostFix 用于传递系统配置工作流程期间最初指定的参数值。

卷名称将从XML文件中提取、并对每个卷执行Ansible克隆攻略手册。



在此示例中、作为实例和中央服务共享同一个卷。因此、只有在使用SAP实例编号时才会执行卷删除 (\$SAPSYSTEM)不是 01。在其他环境中、这可能有所不同、必须相应地进行更改。

```
root@sap-jump:~# cat /usr/sap/scripts/netapp_clone.sh
#!/bin/bash
#Section - Variables
#####
VERSION="Version 0.9"
#Path for ansible play-books
ANSIBLE_PATH=/usr/sap/scripts/ansible
#Values for Ansible Inventory File
PRIMARY_CLUSTER=grenada
PRIMARY_SVM=svm-sap01
PRIMARY_KEYFILE=/usr/sap/scripts/ansible/certs/ontap.key
PRIMARY_CERTFILE=/usr/sap/scripts/ansible/certs/ontap.pem
```

```

#Default Variable if PARAM ClonePostFix / SnapPostFix is not maintained in
LaMa
DefaultPostFix=_clone_1
#TMP Files - used during execution
YAML_TMP=/tmp/inventory_ansible_clone_tmp_$$$.yaml
TMPFILE=/tmp/tmpfile.$$$
MY_NAME="`basename $0`"
BASE_SCRIPT_DIR="`dirname $0`"
#Sendig Script Version and run options to LaMa Log
echo "[DEBUG]: Running Script $MY_NAME $VERSION"
echo "[DEBUG]: $MY_NAME @$@"
#Command declared in the netapp_clone.conf Provider definition
#Command: /usr/sap/scripts/netapp_clone.sh
--HookOperationName=${HookOperationName} --SAPSYSTEMNAME=${SAPSYSTEMNAME}
--SAPSYSTEM=${SAPSYSTEM} --MOUNT_XML_PATH=${MOUNT_XML_PATH}
--PARAM_ClonePostFix=${PARAM-ClonePostFix} --PARAM_SnapPostFix=${PARAM
-SnapPostFix} --PROP_ClonePostFix=${PROP-ClonePostFix}
--PROP_SnapPostFix=${PROP-SnapPostFix}
--SAP_LVM_SRC_SID=${SAP_LVM_SRC_SID}
--SAP_LVM_TARGET_SID=${SAP_LVM_TARGET_SID}
#Reading Input Variables hand over by LaMa
for i in "$@"
do
case $i in
--HookOperationName=*)
HookOperationName="${i#*=}";shift;;
--SAPSYSTEMNAME=*)
SAPSYSTEMNAME="${i#*=}";shift;;
--SAPSYSTEM=*)
SAPSYSTEM="${i#*=}";shift;;
--MOUNT_XML_PATH=*)
MOUNT_XML_PATH="${i#*=}";shift;;
--PARAM_ClonePostFix=*)
PARAM_ClonePostFix="${i#*=}";shift;;
--PARAM_SnapPostFix=*)
PARAM_SnapPostFix="${i#*=}";shift;;
--PROP_ClonePostFix=*)
PROP_ClonePostFix="${i#*=}";shift;;
--PROP_SnapPostFix=*)
PROP_SnapPostFix="${i#*=}";shift;;
--SAP_LVM_SRC_SID=*)
SAP_LVM_SRC_SID="${i#*=}";shift;;
--SAP_LVM_TARGET_SID=*)
SAP_LVM_TARGET_SID="${i#*=}";shift;;
*)
# unknown option

```

```

;;
esac
done
#If Parameters not provided by the User - defaulting to DefaultPostFix
if [ -z $PARAM_ClonePostFix ]; then PARAM_ClonePostFix=$DefaultPostFix;fi
if [ -z $PARAM_SnapPostFix ]; then PARAM_SnapPostFix=$DefaultPostFix;fi
#Section - Functions
#####
#Function Create (Inventory) YML File
#####
create_yml_file()
{
echo "ontapervers:">$YAML_TMP
echo " hosts:">>$YAML_TMP
echo "  ${PRIMARY_CLUSTER}:">>$YAML_TMP
echo "  ansible_host: "'"$PRIMARY_CLUSTER"'>>$YAML_TMP
echo "  keyfile: "'"$PRIMARY_KEYFILE"'>>$YAML_TMP
echo "  certfile: "'"$PRIMARY_CERTFILE"'>>$YAML_TMP
echo "  svmname: "'"$PRIMARY_SVM"'>>$YAML_TMP
echo "  datavolumename: "'"$datavolumename"'>>$YAML_TMP
echo "  snapshotpostfix: "'"$snapshotpostfix"'>>$YAML_TMP
echo "  clonepostfix: "'"$clonepostfix"'>>$YAML_TMP
}
#Function run ansible-playbook
#####
run_ansible_playbook()
{
echo "[DEBUG]: Running ansible playbook
netapp_lama_${HookOperationName}.yml on Volume $datavolumename"
ansible-playbook -i $YAML_TMP
$ANSIBLE_PATH/netapp_lama_${HookOperationName}.yml
}
#Section - Main
#####
#HookOperationName - CloneVolumes
#####
if [ $HookOperationName = CloneVolumes ] ;then
#save mount xml for later usage - used in Section FinalizeCloneVolues to
generate the mountpoints
echo "[DEBUG]: saving mount config..."
cp $MOUNT_XML_PATH /tmp/mount_config_${SAPSYSTEMNAME}_${SAPSYSTEM}.xml
#Instance 00 + 01 share the same volumes - clone needs to be done once
if [ $SAPSYSTEM != 01 ]; then
#generating Volume List - assuming usage of qtrees - "IP-
Adress:/VolumeName/qtrees"
xmlFile=/tmp/mount_config_${SAPSYSTEMNAME}_${SAPSYSTEM}.xml

```

```

if [ -e $TMPFILE ];then rm $TMPFILE;fi
numMounts=`xml_grep --count "/mountconfig/mount" $xmlFile | grep "total: "
| awk '{ print $2 }'`
i=1
while [ $i -le $numMounts ]; do
    xmllint --xpath "/mountconfig/mount[$i]/exportpath/text()" $xmlFile
|awk -F"/" '{print $2}' >>$TMPFILE
i=$((i + 1))
done
DATAVOLUMES=`cat $TMPFILE |sort -u`
#Create yml file and rund playbook for each volume
for I in $DATAVOLUMES; do
datavolumename="$I"
snapshotpostfix="$PARAM_SnapPostFix"
clonepostfix="$PARAM_ClonePostFix"
create_yml_file
run_ansible_playbook
done
else
echo "[DEBUG]: Doing nothing .... Volume cloned in different Task"
fi
fi
#HookOperationName - PostCloneVolumes
#####
if [ $HookOperationName = PostCloneVolumes] ;then
#Reporting Properties back to LaMa Config for Cloned System
echo "[RESULT]:Property:ClonePostFix=$PARAM_ClonePostFix"
echo "[RESULT]:Property:SnapPostFix=$PARAM_SnapPostFix"
#Create MountPoint Config for Cloned Instances and report back to LaMa
according to SAP Note: https://launchpad.support.sap.com/#/notes/1889590
echo "MountDataBegin"
echo '<?xml version="1.0" encoding="UTF-8"?>'
echo "<mountconfig>"
xmlFile=/tmp/mount_config_${SAPSYSTEMNAME}_${SAPSYSTEM}.xml
numMounts=`xml_grep --count "/mountconfig/mount" $xmlFile | grep "total: "
| awk '{ print $2 }'`
i=1
while [ $i -le $numMounts ]; do
MOUNTPOINT=`xmllint --xpath "/mountconfig/mount[$i]/mountpoint/text()"
$xmlFile`;
EXPORTPATH=`xmllint --xpath
"/mountconfig/mount[$i]/exportpath/text()" $xmlFile`;
OPTIONS=`xmllint --xpath "/mountconfig/mount[$i]/options/text()"
$xmlFile`;
#Adopt Exportpath and add Clonepostfix - assuming usage of qtrees - "IP-
Adress:/VolumeName/qtrees"

```

```

TMPFIELD1=`echo $EXPORTPATH|awk -F"/" '{print $1}'`
TMPFIELD2=`echo $EXPORTPATH|awk -F"/" '{print $2}'`
TMPFIELD3=`echo $EXPORTPATH|awk -F"/" '{print $3}'`
EXPORTPATH=$TMPFIELD1"/"$TMPFIELD2$PARAM_ClonePostFix"/"$TMPFIELD3
echo -e '\t<mount fstype="nfs" storagetype="NETFS">'
echo -e "\t\t<mountpoint>${MOUNTPOINT}</mountpoint>"
echo -e "\t\t<exportpath>${EXPORTPATH}</exportpath>"
echo -e "\t\t<options>${OPTIONS}</options>"
echo -e "\t</mount>"
i=$((i + 1))
done
echo "</mountconfig>"
echo "MountDataEnd"
#Finished MountPoint Config
#Cleanup Temporary Files
rm $xmlFile
fi
#HookOperationName - ServiceConfigRemoval
#####
if [ $HookOperationName = ServiceConfigRemoval ] ;then
#Assure that Properties ClonePostFix and SnapPostfix has been configured
through the provisioning process
if [ -z $PROP_ClonePostFix ]; then echo "[ERROR]: Propertiy ClonePostFix
is not handed over - please investigate";exit 5;fi
if [ -z $PROP_SnapPostFix ]; then echo "[ERROR]: Propertiy SnapPostFix is
not handed over - please investigate";exit 5;fi
#Instance 00 + 01 share the same volumes - clone delete needs to be done
once
if [ $SAPSYSTEM != 01 ]; then
#generating Volume List - assuming usage of qtrees - "IP-
Adress:/VolumeName/qtrees"
xmlFile=$MOUNT_XML_PATH
if [ -e $TMPFILE ];then rm $TMPFILE;fi
numMounts=`xml_grep --count "/mountconfig/mount" $xmlFile | grep "total: "
| awk '{ print $2 }'`
i=1
while [ $i -le $numMounts ]; do
    xmllint --xpath "/mountconfig/mount[$i]/exportpath/text()" $xmlFile
|awk -F"/" '{print $2}' >>$TMPFILE
i=$((i + 1))
done
DATAVOLUMES=`cat $TMPFILE |sort -u| awk -F $PROP_ClonePostFix '{ print $1
}'`
#Create yml file and rund playbook for each volume
for I in $DATAVOLUMES; do
datavolumename="$I"

```

```

snapshotpostfix="$PROP_SnapPostFix"
clonepostfix="$PROP_ClonePostFix"
create_yml_file
run_ansible_playbook
done
else
echo "[DEBUG]: Doing nothing .... Volume deleted in different Task"
fi
#Cleanup Temporary Files
rm $xmlFile
fi
#HookOperationName - ClearMountConfig
#####
if [ $HookOperationName = ClearMountConfig ] ;then
    #Assure that Properties ClonePostFix and SnapPostfix has been
    configured through the provisioning process
    if [ -z $PROP_ClonePostFix ]; then echo "[ERROR]: Property
ClonePostFix is not handed over - please investigate";exit 5;fi
    if [ -z $PROP_SnapPostFix ]; then echo "[ERROR]: Property
SnapPostFix is not handed over - please investigate";exit 5;fi
    #Instance 00 + 01 share the same volumes - clone delete needs to
    be done once
    if [ $SAPSYSTEM != 01 ]; then
        #generating Volume List - assuming usage of qtrees - "IP-
        Adress:/VolumeName/qtrees"
        xmlFile=$MOUNT_XML_PATH
        if [ -e $TMPFILE ];then rm $TMPFILE;fi
        numMounts=`xml_grep --count "/mountconfig/mount" $xmlFile
| grep "total: " | awk '{ print $2 }'`
        i=1
        while [ $i -le $numMounts ]; do
            xmllint --xpath
"/mountconfig/mount[$i]/exportpath/text()" $xmlFile |awk -F"/" '{print
$2}' >>$TMPFILE
                i=$((i + 1))
            done
            DATAVOLUMES=`cat $TMPFILE |sort -u| awk -F
$PROP_ClonePostFix '{ print $1 }'`
            #Create yml file and rund playbook for each volume
            for I in $DATAVOLUMES; do
                datavolumename="$I"
                snapshotpostfix="$PROP_SnapPostFix"
                clonepostfix="$PROP_ClonePostFix"
                create_yml_file
                run_ansible_playbook
            done
        done
    fi
fi

```

```
    else
        echo "[DEBUG]: Doing nothing .... Volume deleted in
different Task"
    fi
    #Cleanup Temporary Files
    rm $xmlFile
fi
#Cleanup
#####
#Cleanup Temporary Files
if [ -e $TMPFILE ];then rm $TMPFILE;fi
if [ -e $YAML_TMP ];then rm $YAML_TMP;fi
exit 0
```

## Ansible攻略手册NetApp\_Lam\_CloneVolumes.yml

在Lama系统克隆工作流的CloneVolumes步骤中执行的攻略手册是的组合 `create_snapshot.yml` 和 `create_clone.yml` (请参见 "[NetApp Ansible模块—YAML文件](#)")。本攻略手册可轻松扩展、以涵盖从二级克隆操作和克隆拆分操作等其他使用情形。

```

root@sap-jump:~# cat /usr/sap/scripts/ansible/netapp_lama_CloneVolumes.yml
---
- hosts: ontapserver
  connection: local
  collections:
    - netapp.ontap
  gather_facts: false
  name: netapp_lama_CloneVolumes
  tasks:
  - name: Create SnapShot
    na_ontap_snapshot:
      state: present
      snapshot: "{{ datavolumename }}{{ snapshotpostfix }}"
      use_rest: always
      volume: "{{ datavolumename }}"
      vsserver: "{{ svmname }}"
      hostname: "{{ inventory_hostname }}"
      cert_filepath: "{{ certfile }}"
      key_filepath: "{{ keyfile }}"
      https: true
      validate_certs: false
  - name: Clone Volume
    na_ontap_volume_clone:
      state: present
      name: "{{ datavolumename }}{{ clonepostfix }}"
      use_rest: always
      vsserver: "{{ svmname }}"
      junction_path: '/{{ datavolumename }}{{ clonepostfix }}'
      parent_volume: "{{ datavolumename }}"
      parent_snapshot: "{{ datavolumename }}{{ snapshotpostfix }}"
      hostname: "{{ inventory_hostname }}"
      cert_filepath: "{{ certfile }}"
      key_filepath: "{{ keyfile }}"
      https: true
      validate_certs: false

```

## Ansible攻略手册NetApp\_Lam\_ServiceConfigRemove.yml

在中执行的攻略手册 ServiceConfigRemoval Lama系统销毁工作流的阶段是的组合 delete\_clone.yml 和 delete\_snapshot.yml (请参见 "NetApp Ansible模块—YAML文件")。它必须与的执行步骤一致 netapp\_lama\_CloneVolumes 攻略手册。

```

root@sap-jump:~# cat
/usr/sap/scripts/ansible/netapp_lama_ServiceConfigRemoval.yml
---
- hosts: ontapservers
  connection: local
  collections:
    - netapp.ontap
  gather_facts: false
  name: netapp_lama_ServiceConfigRemoval
  tasks:
    - name: Delete Clone
      na_ontap_volume:
        state: absent
        name: "{{ datavolumename }}{{ clonepostfix }}"
        use_rest: always
        vsserver: "{{ svmname }}"
        wait_for_completion: True
        hostname: "{{ inventory_hostname }}"
        cert_filepath: "{{ certfile }}"
        key_filepath: "{{ keyfile }}"
        https: true
        validate_certs: false
    - name: Delete SnapShot
      na_ontap_snapshot:
        state: absent
        snapshot: "{{ datavolumename }}{{ snapshotpostfix }}"
        use_rest: always
        volume: "{{ datavolumename }}"
        vsserver: "{{ svmname }}"
        hostname: "{{ inventory_hostname }}"
        cert_filepath: "{{ certfile }}"
        key_filepath: "{{ keyfile }}"
        https: true
        validate_certs: false
root@sap-jump:~#

```

## Ansible攻略手册NetApp\_Lama\_ClearMountConfig.yml

攻略手册、在中执行 netapp\_lama\_ClearMountConfig Lama系统刷新工作流的阶段是的组合 delete\_clone.yml 和 delete\_snapshot.yml (请参见 ["NetApp Ansible模块—YAML文件"](#))。它必须与的执行步骤一致 netapp\_lama\_CloneVolumes 攻略手册。

```

root@sap-jump:~# cat
/usr/sap/scripts/ansible/netapp_lama_ServiceConfigRemoval.yml
---
- hosts: ontapservers
  connection: local
  collections:
    - netapp.ontap
  gather_facts: false
  name: netapp_lama_ServiceConfigRemoval
  tasks:
    - name: Delete Clone
      na_ontap_volume:
        state: absent
        name: "{{ datavolumename }}{{ clonepostfix }}"
        use_rest: always
        vsserver: "{{ svmname }}"
        wait_for_completion: True
        hostname: "{{ inventory_hostname }}"
        cert_filepath: "{{ certfile }}"
        key_filepath: "{{ keyfile }}"
        https: true
        validate_certs: false
    - name: Delete SnapShot
      na_ontap_snapshot:
        state: absent
        snapshot: "{{ datavolumename }}{{ snapshotpostfix }}"
        use_rest: always
        volume: "{{ datavolumename }}"
        vsserver: "{{ svmname }}"
        hostname: "{{ inventory_hostname }}"
        cert_filepath: "{{ certfile }}"
        key_filepath: "{{ keyfile }}"
        https: true
        validate_certs: false
root@sap-jump:~#

```

## Ansible inventory.yml示例

此清单文件是在工作流执行期间动态构建的、此处仅显示用于说明。

```
ontapservers:
  hosts:
    grenada:
      ansible_host: "grenada"
      keyfile: "/usr/sap/scripts/ansible/certs/ontap.key"
      certfile: "/usr/sap/scripts/ansible/certs/ontap.pem"
      svmname: "svm-sap01"
      datavolumename: "HN9_sap"
      snapshotpostfix: " _snap_20221115"
      clonepostfix: " _clone_20221115"
```

## 结论

通过将Ansible等现代自动化框架集成到SAP Lama配置工作流程中、客户可以灵活地使用解决方案 来满足标准或更复杂的基础架构要求。

### 从何处查找追加信息

要了解有关本文档中所述信息的更多信息，请查看以下文档和 / 或网站：

- NetApp命名空间中的集合

["https://docs.ansible.com/ansible/latest/collections/netapp/index.html"](https://docs.ansible.com/ansible/latest/collections/netapp/index.html)

- 有关Ansible集成和示例Ansible攻略手册的文档

["https://github.com/sap-linuxlab/demo.netapp\\_ontap"](https://github.com/sap-linuxlab/demo.netapp_ontap)

- Ansible与NetApp的全面集成

["https://www.ansible.com/integrations/infrastructure/netapp"](https://www.ansible.com/integrations/infrastructure/netapp)

- 有关将SAP Lama与Ansible集成的博客

["https://blogs.sap.com/2020/06/08/outgoing-api-calls-from-sap-landscape-management-lama-with-automation-studio/"](https://blogs.sap.com/2020/06/08/outgoing-api-calls-from-sap-landscape-management-lama-with-automation-studio/)

- SAP Landscape Management 3.0企业版文档

["https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/4df88a8f418c5059e1000000a42189c.html#loio4df88a8f418c5059e1000000a42189c"](https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/4df88a8f418c5059e1000000a42189c.html#loio4df88a8f418c5059e1000000a42189c)

- SAP Lama文档—提供商定义

["https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/bf6b3e43340a4cbcb0c0f3089715c068.html"](https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/bf6b3e43340a4cbcb0c0f3089715c068.html)

- SAP Lama文档—自定义挂钩

["https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/139eca2f925e48738a20dbf0b56674c5.html"](https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/139eca2f925e48738a20dbf0b56674c5.html)

- SAP Lama文档—配置SAP Host Agent注册脚本

["https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/250dfc5eef4047a38bab466c295d3a49.html"](https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/250dfc5eef4047a38bab466c295d3a49.html)

- SAP Lama文档—自定义操作和自定义挂钩的参数

["https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/0148e495174943de8c1c3ee1b7c9cc65.html"](https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/0148e495174943de8c1c3ee1b7c9cc65.html)

- SAP Lama文档—自适应设计

["https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/737a99e86f8743bdb8d1f6cf4b862c79.html"](https://help.sap.com/doc/700f9a7e52c7497cad37f7c46023b7ff/3.0.11.0/en-US/737a99e86f8743bdb8d1f6cf4b862c79.html)

- NetApp 产品文档

["https://www.netapp.com/support-and-training/documentation/"](https://www.netapp.com/support-and-training/documentation/)

## 版本历史记录

version	Date	文档版本历史记录
版本 1.0	2023年1月	初始版本。

## 版权信息

版权所有 © 2025 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

## 商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。