

从运维工具 到 workflow 编排

Pilot 驱动的多 Skill 协同架构

一句话需求 → 自动调度 → 审批 → 执行 → 回滚

10 包
完整 Skill 家族

191 工具
MCP 工具总数

14 模板
内置工作流

v1.5.2 · 内部技术分享

Pilot 编排 + 7 个领域 Skill + 统一策略层

vmware-pilot

编排大脑 · 14 工作流模板 · 审批门控

一句话 → 自动调度多 Skill → 审批 → 执行

7 个领域 Skill

aiops(34) · nsx(32) · aria(27) · nsx-security(20)

vks(20) · storage(11) · monitor(7)

vmware-policy

统一策略引擎 · @vmware_tool 装饰器

审计 + 风险分级 + 防注入 · 共享底座

全平台适配

Claude · Codex · Gemini · Cursor · Ollama

本地模型 · MCP 标准协议

运维团队的一天 — 重复、低效、高风险

80%

时间

花在重复操作上

每天循环：巡检 → 查状态 → 做快照 → 收集日志
同样的步骤，每天都要手动执行一遍

5+

界面

来回切换才能完成一项任务

主机状态、告警、性能、存储.....分布在不同页面
每次操作都要在多个 UI 之间反复跳转

0

分级

权限管理粗放，风险无法精确控制

要么全部权限，要么没有权限
缺少只读 / 读写 / 审批的细粒度分级管理

AI 能解决这些问题吗？

能。关键不是「买更好的运维工具」，
而是让 AI 成为你的运维领航员——说人话，管全局，自动编排，全程审计。

「扩容数据库并验证健康」

→ Pilot 自动调度 3 个 Skill
→ Aria 评估 → AIOps 执行 → Monitor 验证
→ 全程审计 + 审批门控

三层架构 — Pilot 驱动的多 Skill 协同

AI Agent （ Claude / Codex / Gemini / Ollama / 本地模型）

用户用自然语言描述需求， Agent 解析意图并转发给 Pilot

vmware-pilot （编排层）

14 内置模板

自定义 workflow

审批门控

状态持久化

自动回滚

▼ 调度 7 个领域 Skill

aiops

34 工具

nsx

32 工具

aria

27 工具

nsx-sec

20 工具

vks

20 工具

storage

11 工具

monitor

7 工具

▼ 每个工具经过 @vmware_tool 装饰器

vmware-policy （共享底座）

- @vmware_tool 装饰器
- 统一审计 audit.db
- 风险分级 R1-R4
- 策略引擎
- 防注入检查

10 个包 · 191 个 MCP 工具 · 统一架构

Package	MCP 工具	定位	核心能力
vmware-pilot	11	编排器	工作流编排 · 审批 · 状态管理 · 14 模板
vmware-aiops	34	全能运维	VM 生命周期 · 快照 · 迁移 · 部署 · Guest 操作
vmware-nsx	32	网络管理	Segment · Gateway · NAT · 路由 · IPAM
vmware-aria	27	可观测性	指标 · 告警 · 容量 · 异常 · LLM 报表
vmware-nsx-security	20	微分段安全	DFW 策略 · 安全组 · IDPS · Traceflow
vmware-vks	20	K8s 管理	Supervisor · TKC 生命周期 · Namespace
vmware-storage	11	存储管理	Datastore · iSCSI · vSAN 健康与容量
vmware-monitor	7	只读监控	安全只读 · 清单查询 · 告警 · 扫描
vmware-policy	—	共享底座	@vmware_tool · 审计 · 策略 · 防注入

统一模式：所有 191 个工具使用 @vmware_tool 装饰器 → 统一审计 + 策略引擎 + 防注入 + MCP 标准 annotations 。 policy 无 MCP 工具，是共享库。

vmware-pilot — 为什么需要编排层？

Pilot 解决了什么？

- ① **多 skill 协同**
一个需求需要跨多个 skill 执行，手动串联效率低
- ② **审批门控**
写操作自动触发审批流程，非管理员不可越级执行
- ③ **状态持久化**
 workflow 执行状态持久保存，断点续做，支持回滚
- ④ **自定义扩展**
YAML 定义 + AI 辅助设计 + create_workflow 即时注册

示例：「扩容数据库集群」

用户说	「帮我扩容 db 集群到 8 核 32G」
Pilot	解析意图 → 选择 capacity_expansion 模板
Aria	评估当前容量 + 预测增长趋势
AIOps	执行 VM reconfigure（审批后）
Monitor	验证扩容后健康状态 + 性能指标
Pilot	汇总报告 → 标记完成 / 异常回滚

Pilot 的 11 个 MCP 工具

plan_workflow

run_workflow

get_workflow_status

approve

rollback

list_workflows

create_workflow

design_workflow

update_draft

confirm_draft

get_skill_catalog

14 个内置模板 — 四大类开箱即用 + 自定义扩展

变更管理

clone_and_test

plan_and_approve

rolling_restart

capacity_expansion

patch_deployment

5 ↑

基础设施部署

network_segment_setup

vks_cluster_deploy

storage_expansion

3 ↑

运维保障

incident_response

disaster_recovery

compliance_scan

3 ↑

基线管理

baseline_capture

baseline_audit

baseline_remediate

3 ↑

自定义工作流

YAML 定义 手写 YAML 工作流配置 →
create_workflow 注册

AI 辅助设计 design_workflow : 用自然语言描述 → AI 生成 YAML

即时生效 create_workflow 注册后立即可通过
Pilot 调度执行

vmware-aiops — 34 个 MCP 工具 · 全能运维

清单与查询 (8)

- `get_vm_info`
- `list_vms`
- `list_hosts`
- `list_clusters`
- `list_datastores`
- `list_networks`
- `list_resource_pools`
- `list_folders`

生命周期 (9)

- `create_vm`
- `delete_vm`
- `clone_vm`
- `reconfigure_vm`
- `rename_vm`
- `power_on_vm`
- `power_off_vm`
- `reset_vm`
- `suspend_vm`

快照与迁移 (6)

- `create_snapshot`
- `list_snapshots`
- `revert_snapshot`
- `delete_snapshot`
- `migrate_vm`
- `relocate_vm`

Guest 与高级 (11)

- `guest_run_command`
- `guest_upload_file`
- `guest_download_file`
- `guest_list_processes`
- `deploy_ovf`
- `list_events`
- `list_alarms`
- `get_alarm_status`
- `acknowledge_alarm`
- `run_security_scan`
- `get_scan_results`

vmware-monitor — 7 个 MCP 工具 · 安全只读

7 个只读 MCP 工具

`list_vms` 虚拟机清单 + 状态 + 资源

`list_hosts` ESXi 主机清单 + 健康状态

`get_vm_info` 单个 VM 详细信息

`list_alarms` 活跃告警列表 + 严重级别

`list_events` 事件日志 + 时间范围过滤

`run_security_scan` 42 项安全检查 + 合规报告

`get_scan_results` 历史扫描结果查询

为什么单独拆出 Monitor ?

最小权限原则

只读场景不暴露写操作工具

减少 AI 误操作风险面

轻量快速

7 个工具 vs aiops 的 34 个

加载快、上下文少、响应准

分级授权

初级运维 → monitor

高级运维 → aiops

安全审计 → monitor + aria

设计理念： Monitor 是 aiops 的安全子集。给 AI Agent 只装 monitor = 只读巡检员，零写操作风险。

vmware-nsx (32) + nsx-security (20) — 网络与微分段

vmware-nsx · 32 工具 · 网络管理

Segment 管理 (6)

```
list_segments · get_segment · create_segment ·  
update_segment · delete_segment · list_segment_ports
```

Gateway & NAT (6)

```
list_gateways · get_gateway · list_nat_rules ·  
create_nat_rule · delete_nat_rule · list_lb_services
```

路由 & BGP (5)

```
list_routing_table · get_bgp_neighbors ·  
list_route_maps · list_prefix_lists ·  
get_routing_config
```

IPAM & DHCP (5)

```
list_ip_pools · allocate_ip · release_ip ·  
list_dhcp_servers · get_ip_pool_usage
```

监控 & 拓扑 (6)

```
get_nsx_status · list_transport_nodes ·  
get_edge_cluster · list_logical_switches ·  
get_nsx_alarms · get_network_topology
```

vmware-nsx-security · 20 工具 · 微分段

DFW 策略 & 规则 (7)

```
list_dfw_policies · get_dfw_policy · create_dfw_policy ·  
create_dfw_rule · update_dfw_rule · delete_dfw_rule ·  
reorder_dfw_rule
```

安全组 & 标签 (6)

```
list_security_groups · create_security_group ·  
list_vm_tags · tag_vm · untag_vm · get_effective_rules
```

Traceflow & IDPS (6)

```
start_traceflow · get_traceflow_result ·  
list_idps_profiles · get_idps_status ·  
update_idps_profile · get_idps_events
```

拆分原则： nsx = 网络基础设施（Segment/Gateway/路由）， nsx-security = 安全策略（DFW/安全组/IDPS）。不同团队可分别授权。

vmware-aria — 27 个 MCP 工具 · 可观测性 + LLM 报表

Aria 27 个工具 — 五大能力域	
指标查询 (7)	query_vm_metrics · query_host_metrics · query_cluster_metrics
告警管理 (6)	list_metric_keys · get_metric_history · compare_metrics · get_baseline
容量规划 (5)	list_alerts · get_alert · acknowledge_alert · close_alert
异常检测 (5)	list_alert_definitions · get_alert_trend · get_capacity_remaining · forecast_capacity · what_if_analysis
报表生成 (4)	get_reclaimable · optimize_placement · detect_anomalies · get_anomaly_history · list_anomaly_definitions
	correlate_anomalies · get_root_cause
	generate_report · list_reports · get_report · schedule_report

LLM 生成报表

Aria 采集数据 + LLM 生成自然语言报告

- 1

用户说

「给我出一份上周的容量报告」
- 2

Aria 采集

query_vm_metrics + get_capacity_remaining
- 3

LLM 分析

分析数据趋势，生成自然语言摘要
- 4

输出报告

容量现状 + 风险预警 + 优化建议

Aria + LLM 典型场景

<div>周报自动化</div> <div>每周一自动生成集群健康报告 发送给运维团队 + 管理层</div>	<div>容量预警</div> <div>forecast_capacity 预测 30 天用量 LLM 生成扩容建议 + 时间窗口</div>	<div>异常诊断</div> <div>detect_anomalies + correlate LLM 关联分析根因 + 修复建议</div>	<div>资源优化</div> <div>get_reclaimable 找闲置资源 LLM 生成回收方案 + ROI 估算</div>
---	--	---	--

vmware-vks (20) + vmware-storage (11) — K8s 与存储

vmware-vks · 20 工具 · Tanzu K8s 管理

兼容性检查 (3)

- check_compat_matrix
- validate_config
- get_supported_versions

Namespace 管理 (5)

- list_namespaces
- update_namespace_quota
- get_namespace_status
- create_namespace
- delete_namespace

TKC 生命周期 (7)

- list_tkc
- scale_tkc
- delete_tkc
- create_tkc
- upgrade_tkc
- get_tkc_status

监控 & 运维 (5)

- list_tkc_nodes
- drain_node
- uncordon_node
- get_tkc_config
- get_node_health
- cordon_node

vmware-storage · 11 工具 · 存储管理

Datastore (4)

- list_datastores
- browse_datastore
- get_datastore_capacity
- scan_datastores

iSCSI (4)

- list_iscsi_adapters
- add_iscsi_target
- remove_iscsi_target
- rescan_iscsi

vSAN (3)

- get_vsan_health
- get_vsan_capacity
- list_vsan_disk_groups

拆分逻辑： vks = Tanzu K8s 集群全生命周期， storage = 数据存储（Datastore/iSCSI/vSAN）。两者独立部署，按需安装。

vmware-policy — @vmware_tool 装饰器 · 审计 · 策略引擎

@vmware_tool 装饰器 — 每个工具的必经之路



代码示例

```
@vmware_tool(
    risk_level="R3",
    requires_confirm=True,
    description="创建 VM 快照"
)
async def create_snapshot(
    vm_name: str,
    snapshot_name: str
) -> dict:
    # 实际 API 调用
    ...
```

风险分级 — 从只读到高危

R1 · 只读

查询、列表、状态检查
无需确认，直接执行

list_vms · get_vm_info ·
list_alarms

R2 · 低风险写

快照、标签、备注
单次确认后执行

create_snapshot · tag_vm ·
rename_vm

R3 · 中风险写

VM 配置变更、迁移
需审批 + 确认

reconfigure_vm · migrate_vm
· scale_tkc

R4 · 高危操作

删除、批量变更
多级审批 + Dry-Run

delete_vm · delete_namespace
· bulk_ops

跨 Skill 路由 — 自动识别场景，精准调度

monitor



aiops

Monitor 发现告警

list_alarms → 发现 VM 告警 → 自动切换到 aiops 执行修复

「检查告警」→ 发现 VM CPU 100% → 「帮我扩容这台 VM」

aria



pilot

Aria 数据触发 workflow

容量预测触发扩容模板 → Pilot 编排多 Skill 执行扩容

forecast_capacity → 30 天后超载 → capacity_expansion 模板

nsx



nsx-security

NSX 规则变更需安全审查

create_segment → 需要配套 DFW 规则 → 路由到 security 创建策略

「创建 DMZ 网段」→ 自动关联 → 「配置防火墙规则」

pilot



多 Skill 协同

Pilot 编排全链条

execute_workflow → 依模板步骤依次调度不同 Skill → 汇总结果

「执行安全审计」→ monitor(扫描) + nsx-sec(DFW) + aria(报告)

设计原则： Skill 之间不直接调用。路由由 AI Agent 或 Pilot 模板驱动，每次路由切换都有审计记录。

六层安全体系 — vmware-policy 是安全中枢

L1 Skill 分级

只读 (monitor) vs 全能 (aiops)
最小权限, 按需分配

L2 工具风险标注

@vmware_tool 装饰器
R1 只读 / R2 低危 / R3 中危 / R4 高危

L3 策略引擎

可配置的策略规则
按角色 / 时间 / 环境动态决策

L4 审批门控

Pilot workflow 审批
高风险操作需人工批准

L5 防注入

参数正则扫描
拒绝 SQL/Shell/ 路径注入

L6 全程审计

audit.db 不可篡改记录
操作人 / 时间 / 参数 / 结果

安全中枢: vmware-policy 是所有安全层的代码实现。191 个工具无一例外经过 @vmware_tool 装饰器 → 策略检查 → 审计记录 + MCP annotations 。

9+ AI 平台 · 本地模型 · MCP 标准协议

云端 AI 平台

Claude (Anthropic)	原生 MCP 支持, 推荐首选
OpenAI Codex / GPT	通过 MCP Bridge 接入
Gemini (Google)	MCP 协议兼容
Cursor / Windsurf	IDE 内嵌 MCP 工具
Claude Code (CLI)	终端 + MCP 原生集成

本地部署 (数据不出网)

- Ollama 一键本地运行, Docker 就绪
- VCF + Ollama VMware 私有云 + 本地推理
- vLLM / TGI 企业级推理服务
- 任意 OpenAI 兼容 标准 API 即可接入

MCP (Model Context Protocol) — 为什么选择 MCP ?

标准化工具描述

JSON Schema 定义工具参数和返回值
AI 自动理解工具用途, 无需额外训练

平台无关

同一套工具定义可运行在任意 AI 平台
一次开发, 多平台部署

安全边界清晰

工具在服务端执行, AI 只发送调用请求
天然隔离, 凭证不暴露给 AI

生态开放

Smithery 分发 · OpenClaw 上架
pip/uv 安装 · 插件式扩展

本地模型 + Agent 选型指南

Agent	类型	MCP	本地模型	特点	适合场景
Aider	CLI	间接	Ollama/LM Studio	最简部署，对话式	快速上手
Continue	VS Code	原生	Ollama	IDE 内操作	开发者日常
Goose	CLI Agent	原生	Ollama/vLLM	Block 开源	MCP 自动化
Cline	VS Code	原生	任意兼容	原 Claude Dev	IDE 全能
OpenCode	CLI	原生	75+ 提供商	终端优先	本地 MCP
AnythingLLM	桌面	原生	本地	开箱即用	非技术人员
n8n	工作流	原生	任意 API	可视化编排	自动巡检
Dify	LLMOps	原生	Ollama	Agent+RAG	团队协作
CrewAI	多 Agent	原生	Ollama	角色分工	多步编排
Gemini CLI	CLI	原生	Gemini	Google 开源	长上下文

SE 快速 Demo

Aider + Ollama + Qwen3.5-35B

自动化巡检

n8n + MCP + 任意模型

安全敏感 / 气隙

OpenCode + Ollama + 本地模型

方式一：Aider + Ollama（最快上手）

```
1. ollama pull qwen3.5:35b-a3b
2. pip install aider-chat
3. aider --model ollama/qwen3.5:35b-a3b \
  --conventions codex-skill/AGENTS.md
```

5 分钟部署 · 纯 CLI · 无需配置 MCP

CLI 模式

方式二：Continue + MCP（IDE 原生）

1. 安装 Continue 插件
2. 配置 Ollama 模型 + MCP Server
3. 在 IDE 内直接对话操作 vCenter

MCP 原生 · IDE 内操作 · 结构化工具调用

MCP 模式

模型	参数量	VRAM	准确率	推荐模式	适合
Phi-4 14B	14B	~9GB	~70%	CLI	监控 + 简单
Gemma 4 26B MoE	26B/4B	~12GB	~85%	CLI	日常运维
Qwen3.5-35B-A3B	35B/3B	~25GB	~88%	CLI/MCP	全功能
Llama 4 Scout 109B	109B/17B	~55GB	~92%	MCP	推荐：全功能

写操作建议 MoE 大模型（Llama 4 / Qwen3.5），确保工具调用可靠性

5 分钟安装 · 4 种方式 · 即装即用

pip / uv install

推荐

```
pip install vmware-aiops
vmware-pilot
# 或
uv pip install vmware-aiops
vmware-pilot
```

标准 Python 包安装，适合所有环境

Smithery

一键

```
npx @smithery/cli install \
@zhouwei/vmware-aiops \
--client claude
```

Smithery 平台一键安装到 Claude Desktop

Claude Code

CLI

```
claude mcp add vmware-aiops \
-- uvx vmware-aiops
```

Claude Code CLI 直接注册 MCP Server

手动配置

灵活

```
//
claude_desktop_config.json
"vmware-aiops": {
  "command": "uvx",
  "args": ["vmware-aiops"]
}
```

直接编辑 AI 平台配置文件

安装后：在 AI 对话中直接说「帮我检查所有 VM 状态」→ AI 自动调用 vmware-monitor 工具 → 返回结构化结果。无需学习任何命令。

三步走 — Demo · POC · 交付



VMware AI Skill 家族

从运维工具到 workflow 编排

让 AI 成为 你的运维领航员

10 个 Package

191 个 MCP 工具

14 个工作流模板

6 层安全体系

一句话说清楚

对运维团队

说人话管基础设施，不用学命令，
不用切界面，不用写脚本。

对安全团队

六层安全 + 全程审计 + 风险分级，
AI 操作比人工操作更可控。

对管理层

运维效率提升 10x，一个 Pilot
模板替代一套 SOP 文档。

对架构师

MCP 标准 + 本地部署 + 多平台，
技术选型无锁定风险。

v1.5.2 · 2026 · 开源 · PyPI · Smithery

Contributors: Wei Zhou (zw008) · tangbiao0700 · yjs-2026 — 欢迎加入！