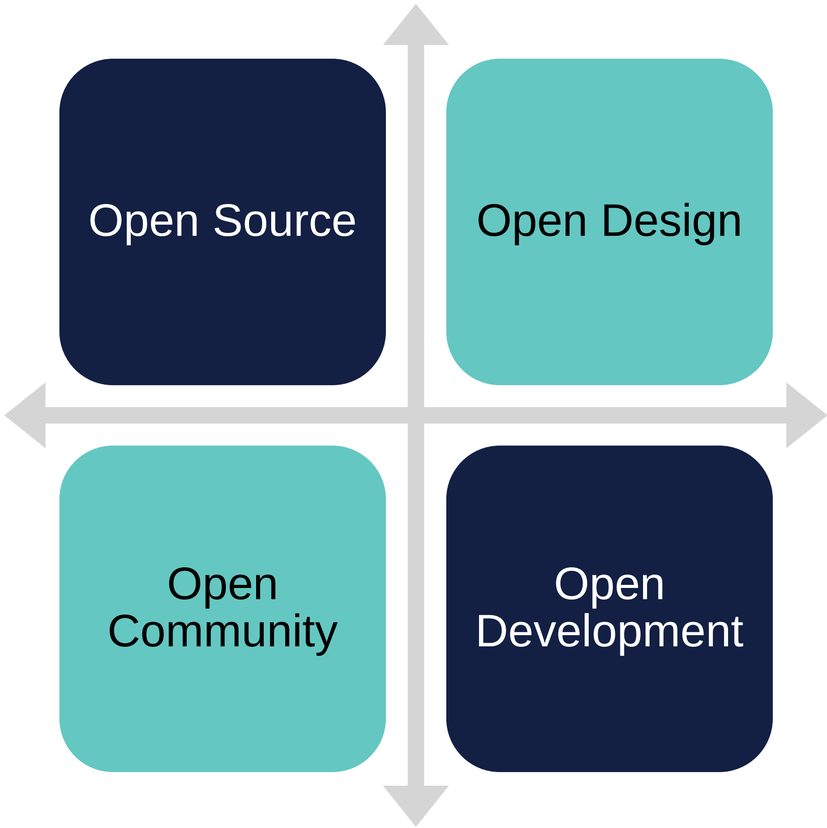
Onboarding | 2020 Project Teams Gathering

airship

New Developer Onboarding

- Airship overview & history
- Airship projects & related projects
- Stepping through an Airship deployment
- Learning more
- Getting involved



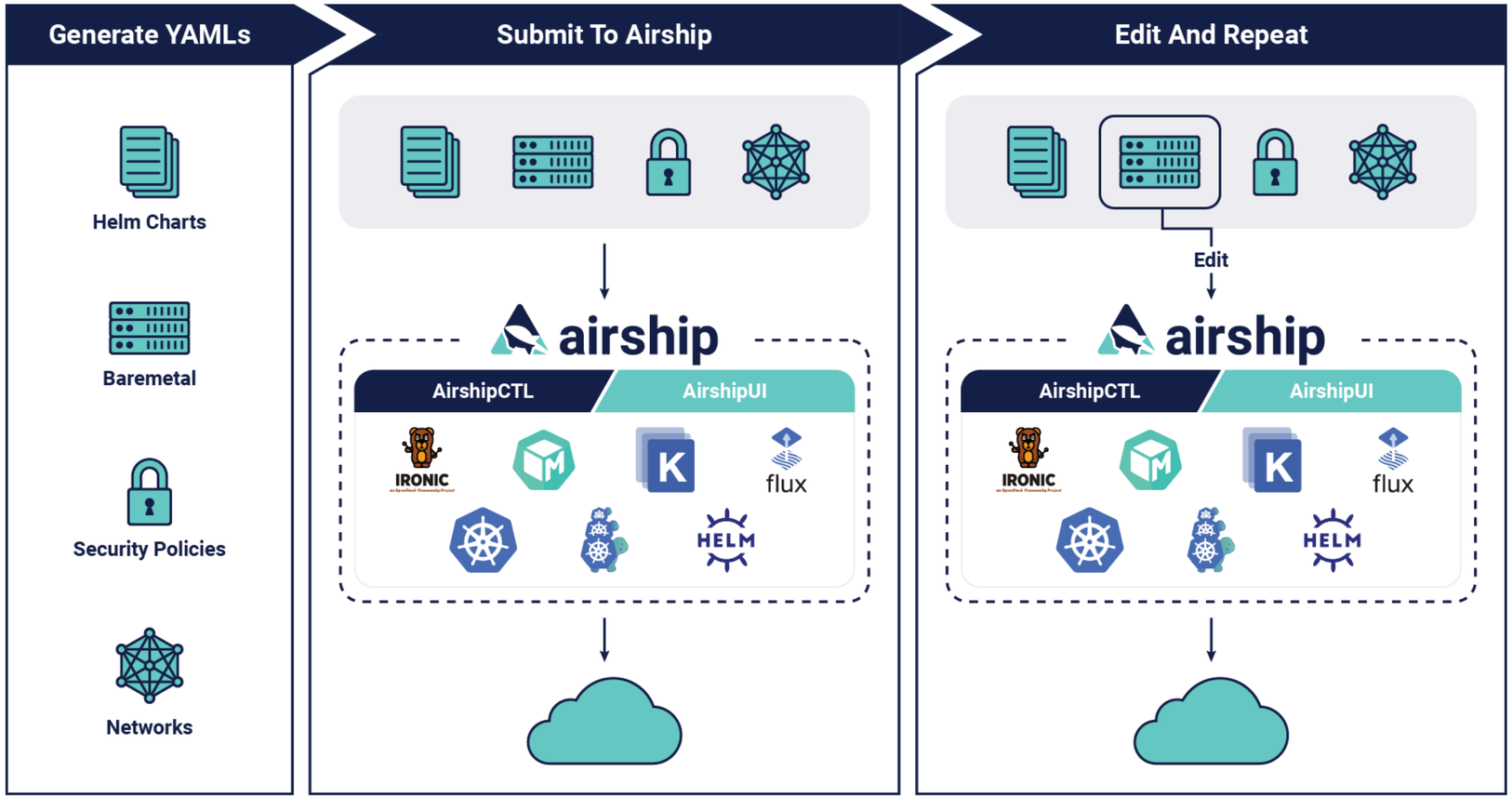
airship

Airship Overview

"Airship is a collection of loosely coupled, but interoperable, open source tools that declaratively automate cloud provisioning."

Motivations

- Orchestration: Batteries included
- Singular Deployment Method
- Predictable Upgrades
- Repeatable Multi-site Deployments
- Resiliency
- Enterprise-Grade Security



airship

History of Airship

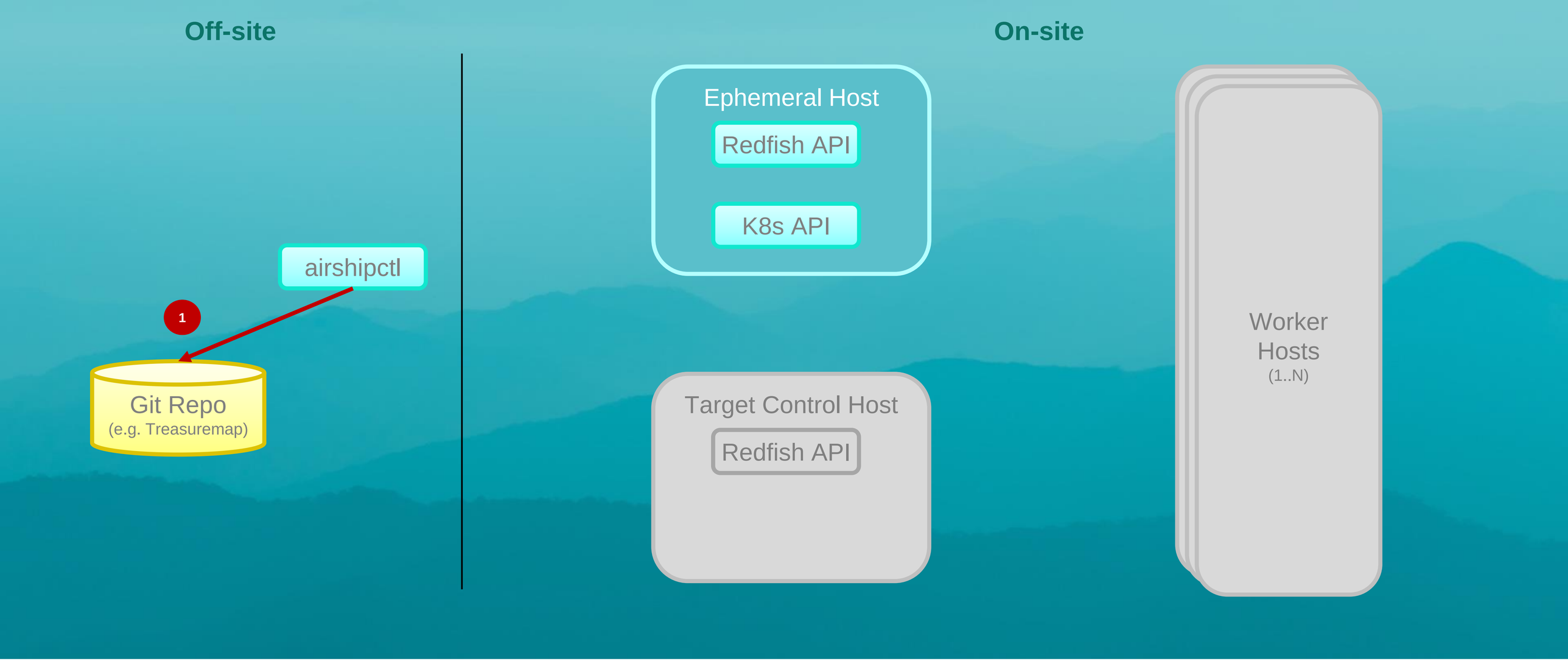
The Evolution to Airship 2.0

- Smaller, ephemeral footprint
- Adoption of established upstream projects
- Less downstream customization of manifests
- Expanded support for multiple platforms (public cloud, Docker, bare metal)
- Improved speed of deployment vs. Airship 1.0
- Airship UI enhances the user experience
- Airship in a Pod improves the onboarding experience
- Support for smaller deployments

airship

Airship 2.0 – Components involved

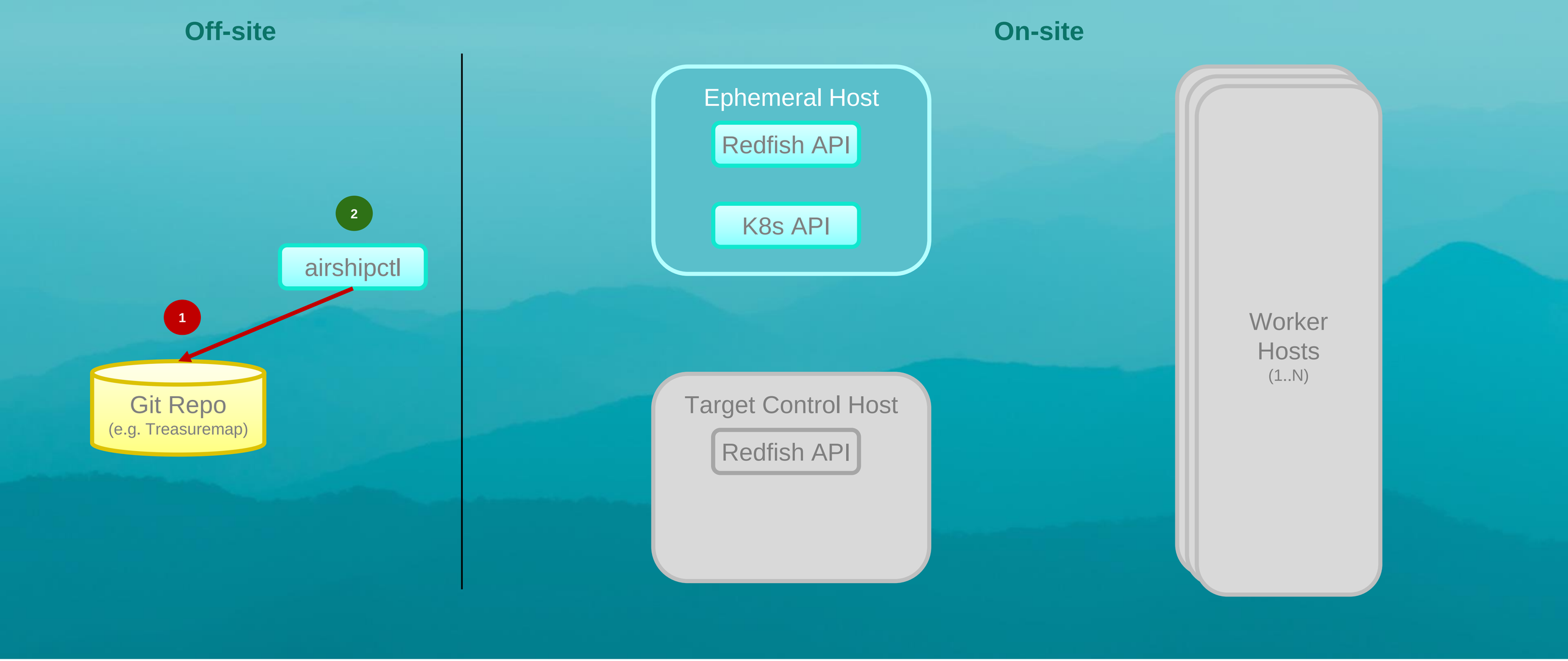
Component	Function
AirshipCTL	Integrating and Orchestrating open infrastructure
Airship UI	Visualization and management of Kubernetes clusters and declarative intent
Airship ImageBuilder	Declarative ISO/QCOW generation
Kustomize	Deduplication of declarative intent at scale
KubeADM	Kubernetes configuration and management
Cluster API	Declarative provisioning of Kubernetes Clusters
Metal3 and Ironic	Declarative provisioning of bare metal infrastructure
Flux Helm Operator	Declarative Helm Chart management
Airship HostConfig Operator	Declarative Day-2 host management
Airship Treasuremap	Library of integrated declarative intent for operators to consume and extend



1 `airshipctl document pull` **fetches YAML repositories from Git**

Step 1:

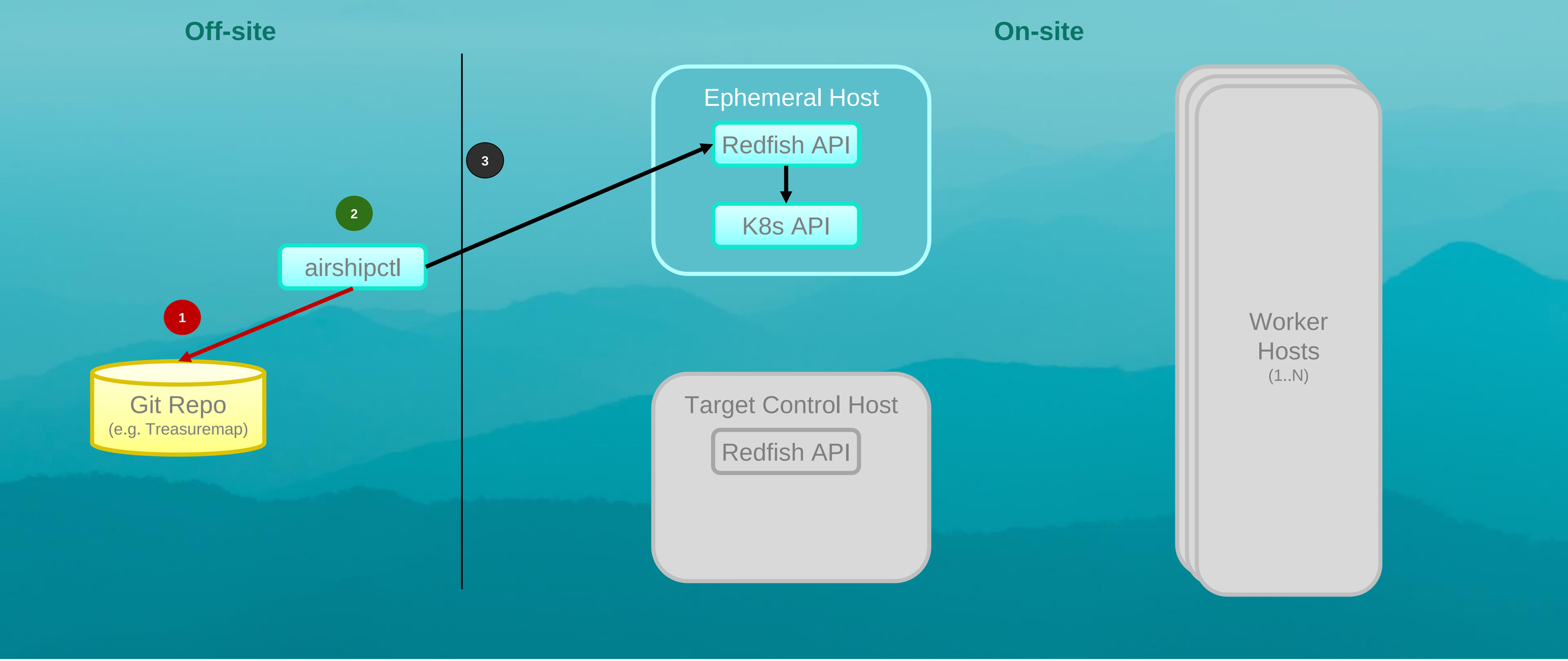
Preparation and Ephemeral Deployment



Step 1:

Preparation and Ephemeral Deployment

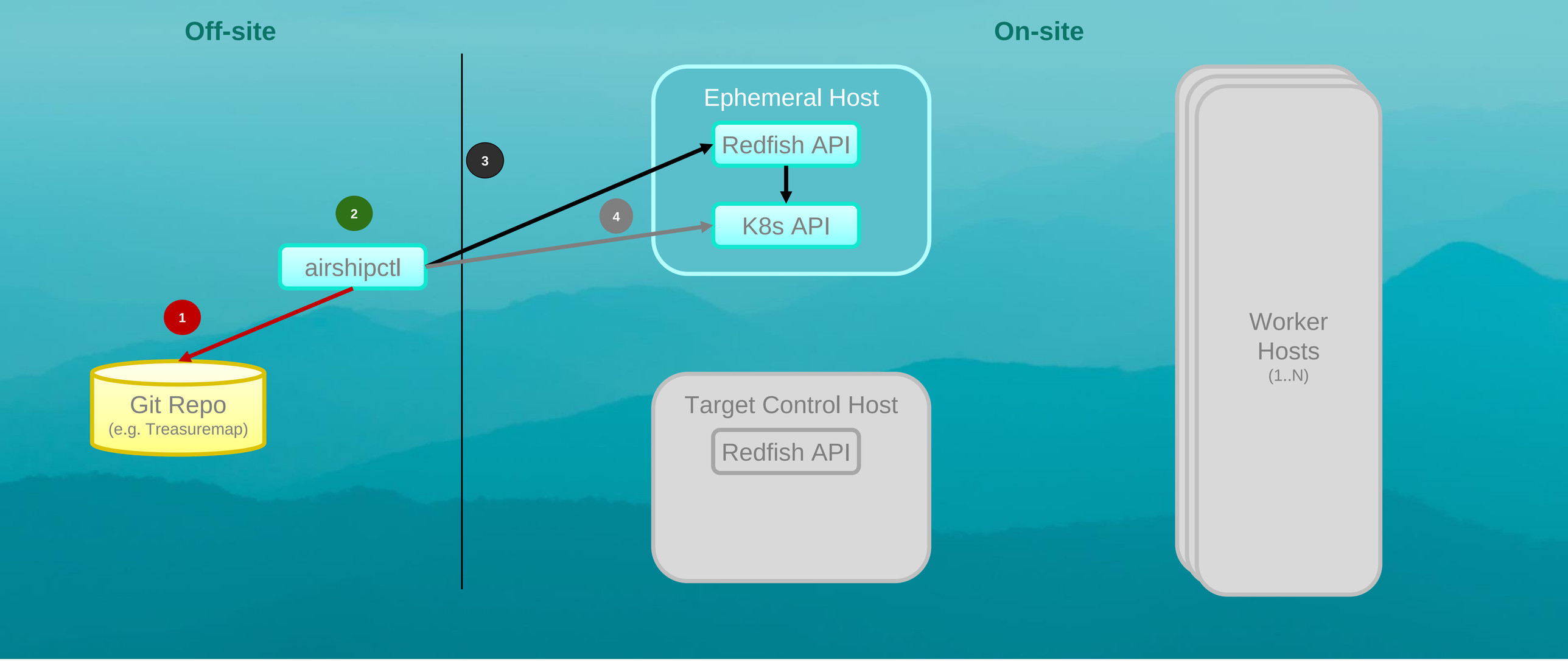
1. `airshipctl document pull` **fetches YAML repositories from Git**
2. `airshipctl phase run bootstrap` **builds an ISO for the ephemeral node**



Step 1:

Preparation and Ephemeral Deployment

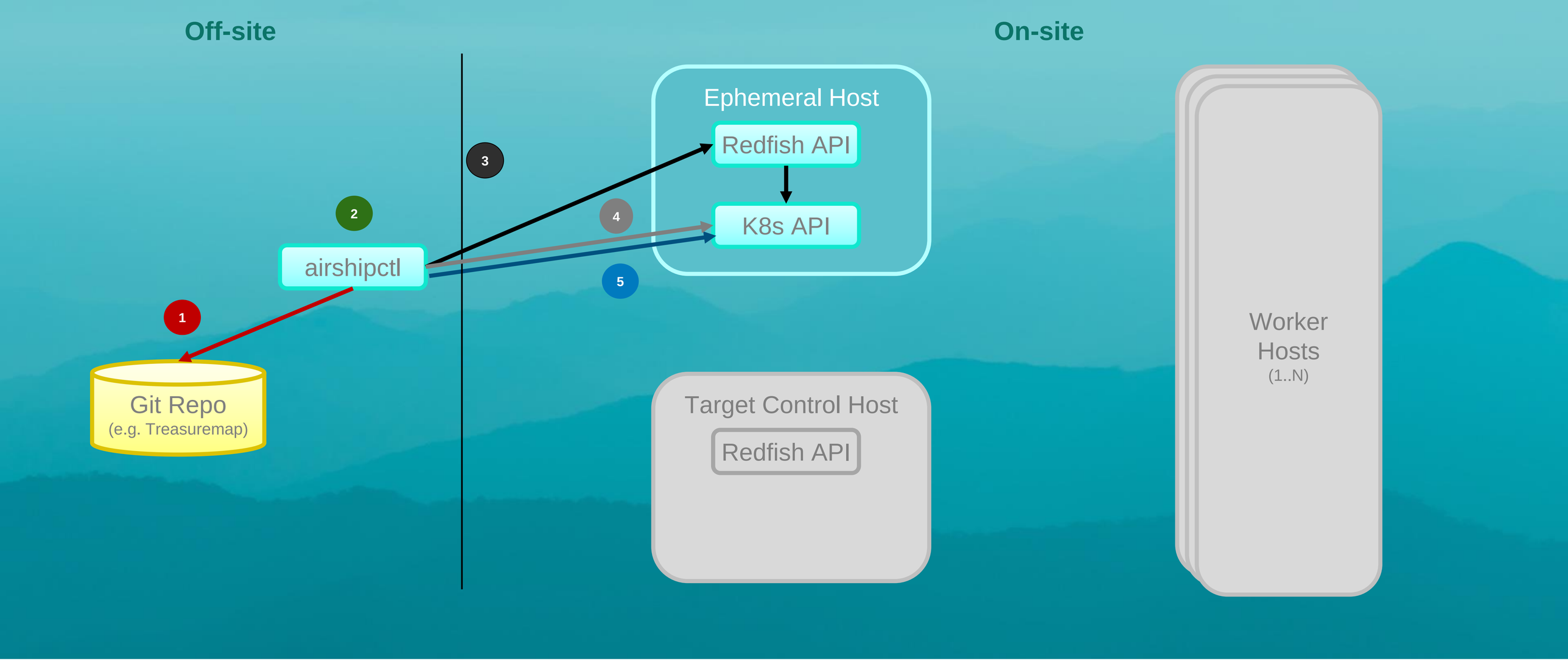
1. `airshipctl document pull` fetches YAML repositories from Git
2. `airshipctl phase run bootstrap` builds an ISO for the ephemeral node
3. `airshipctl baremetal remotedirect` boots the ephemeral node/cluster over redfish



Step 1:

Preparation and Ephemeral Deployment

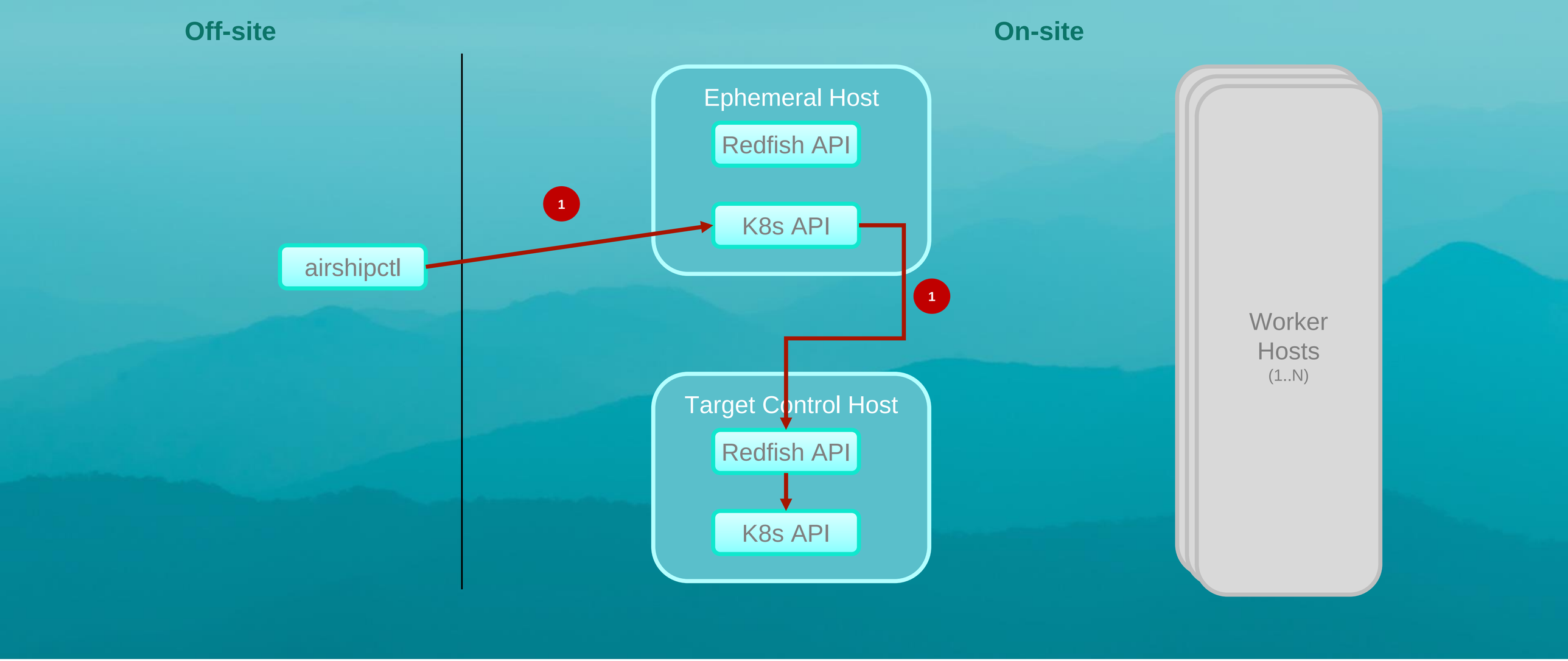
1. `airshipctl document pull` fetches YAML repositories from Git
2. `airshipctl phase run bootstrap` builds an ISO for the ephemeral node
3. `airshipctl baremetal remotedirect` boots the ephemeral node/cluster over redfish
4. `airshipctl phase run initinfra-ephemeral` deploys Metal3 and Calico



Step 1:

Preparation and Ephemeral Deployment

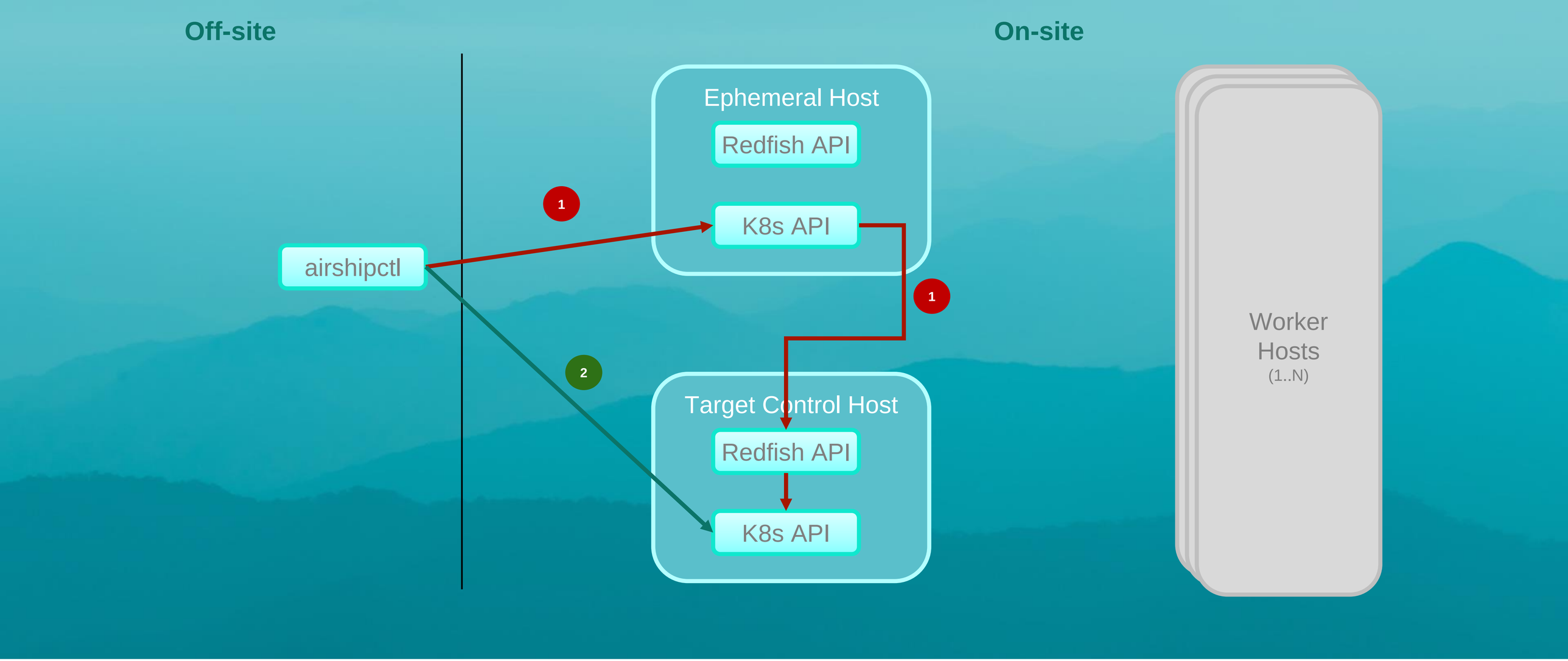
1. `airshipctl document pull` fetches YAML repositories from Git
2. `airshipctl phase run bootstrap` builds an ISO for the ephemeral node
3. `airshipctl baremetal remotedirect` boots the ephemeral node/cluster over redfish
4. `airshipctl phase run initinfra-ephemeral` deploys Metal3 and Calico
5. `airshipctl phase run clusterctl-init-ephemeral` deploys CAPI controllers



1 `airshipctl phase run controlplane-ephemeral` **provisions the target control plane**

Step 2:

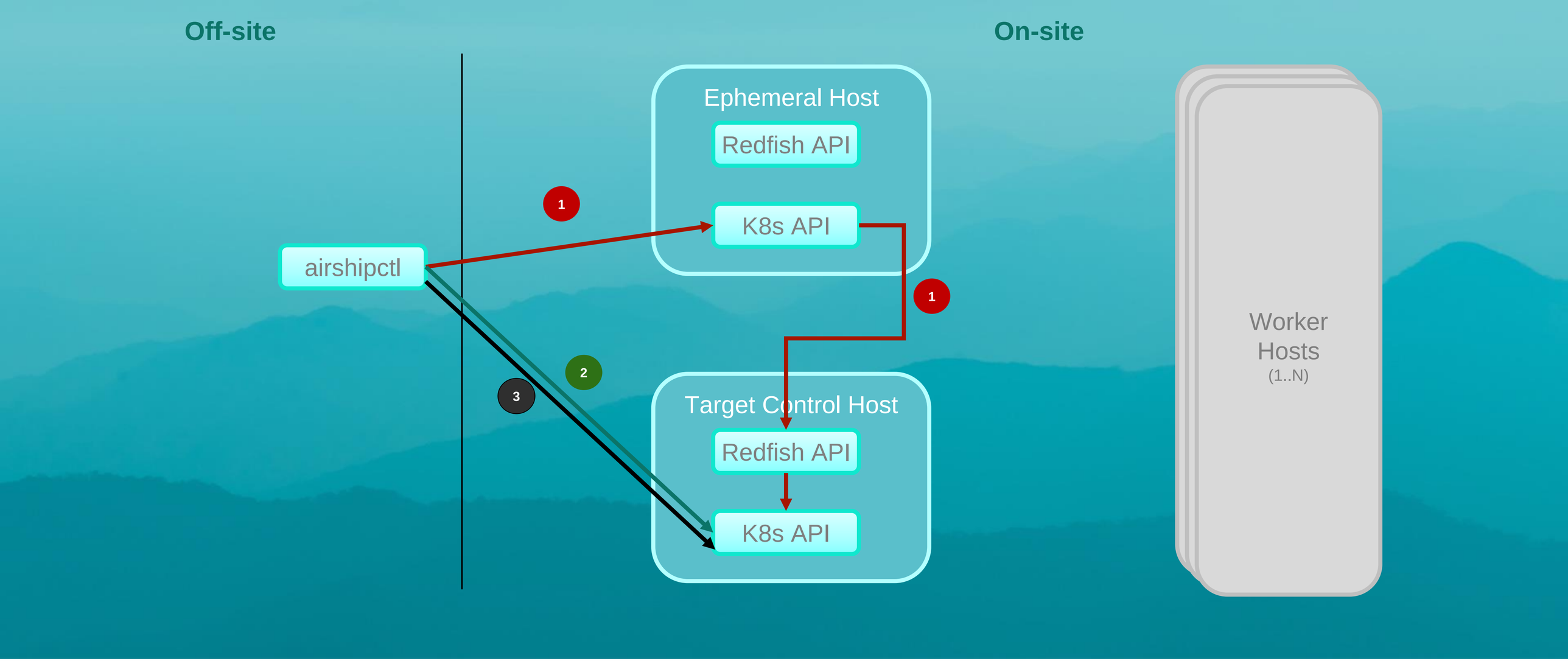
Provision Target Cluster



Step 2:

Provision Target Cluster

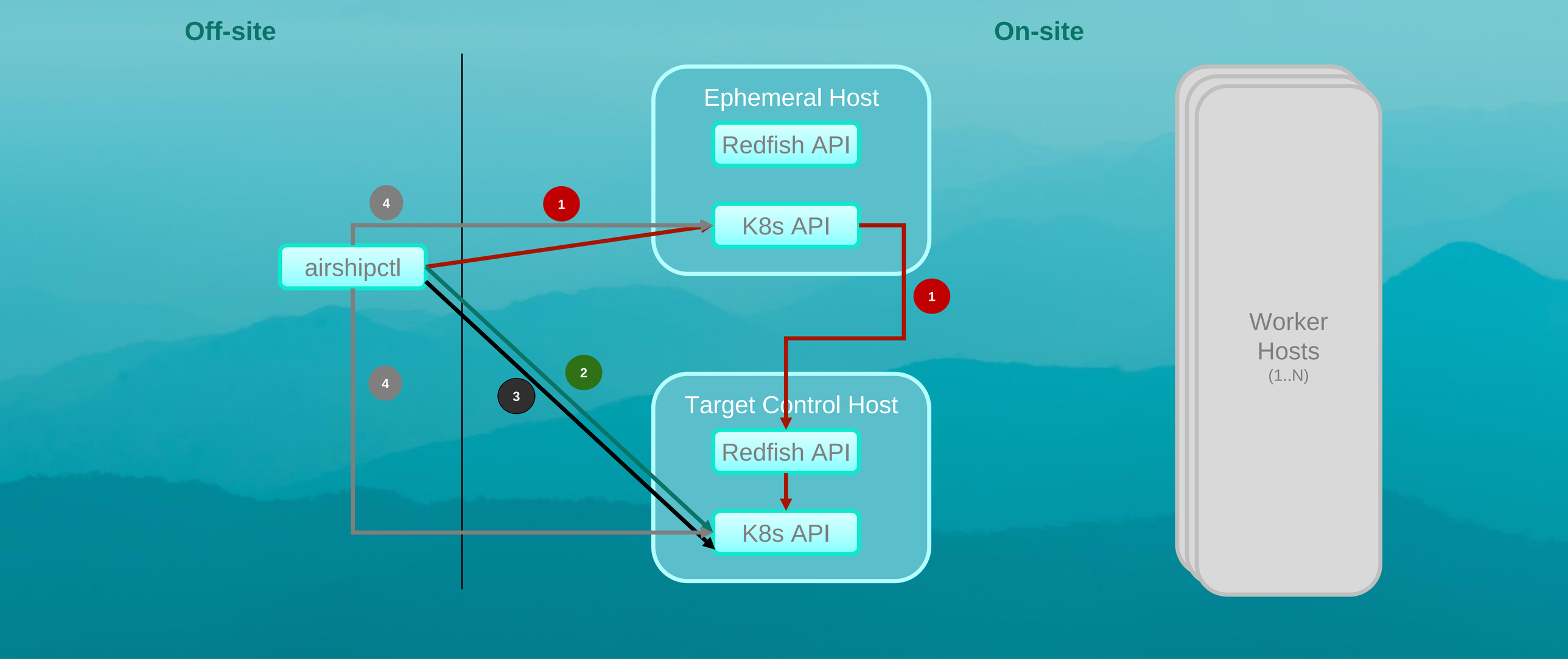
1. `airshipctl phase run controlplane-ephemeral` **provisions the target control plane**
2. `airshipctl phase run initinfra-target` **deploys Metal3, Calico, and Helm-Operator**



Step 2:

Provision Target Cluster

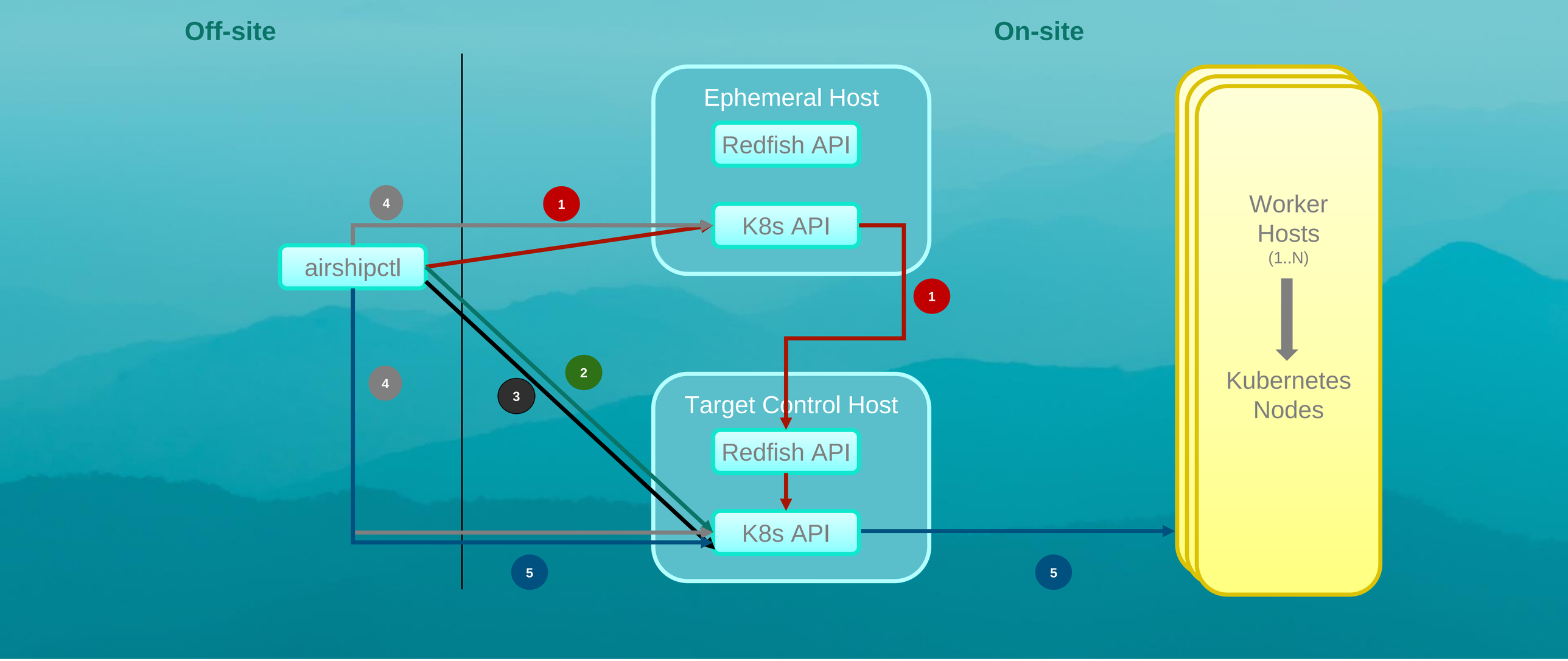
1. `airshipctl phase run controlplane-ephemeral` **provisions the target control plane**
2. `airshipctl phase run initinfra-target` **deploys Metal3, Calico, and Helm-Operator**
3. `airshipctl baremetal clusterctl-init-target` **deploys CAPI controllers**



Step 2:

Provision Target Cluster

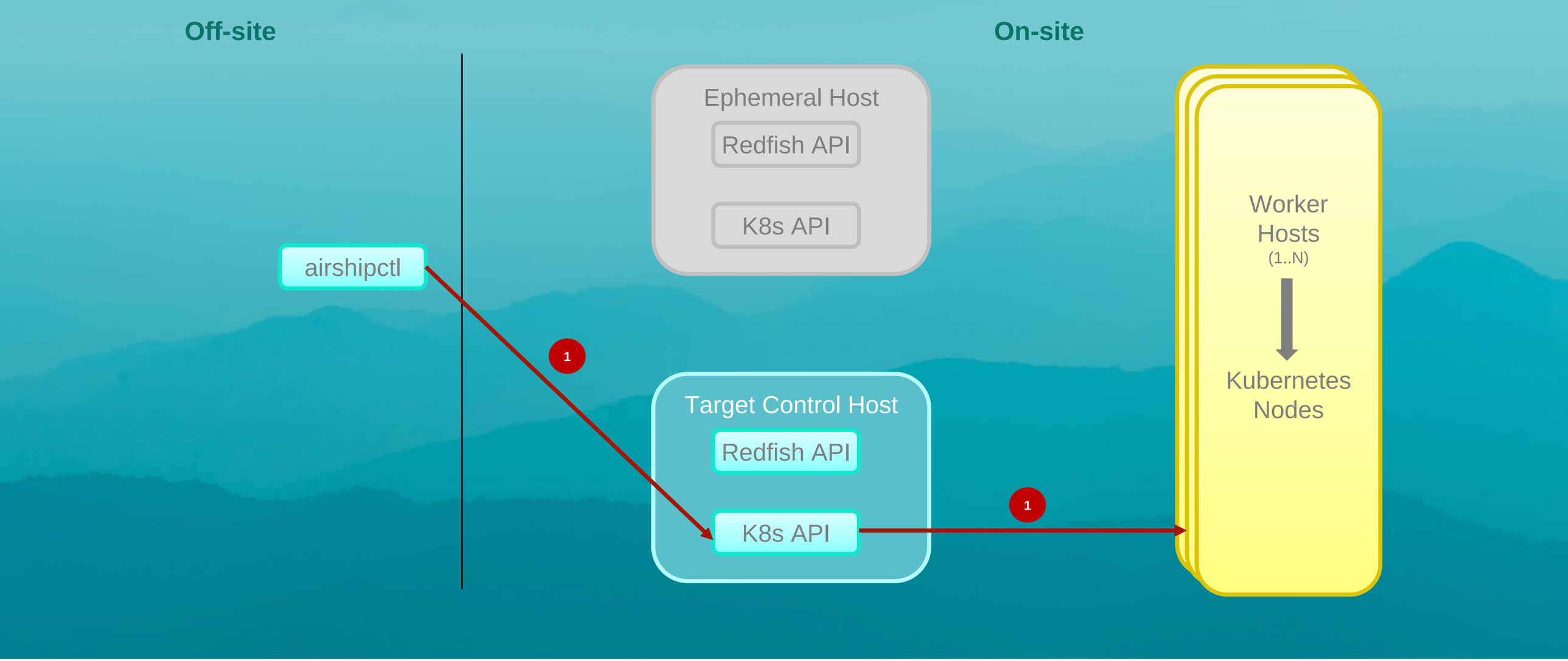
1. `airshipctl phase run controlplane-ephemeral` provisions the target control plane
2. `airshipctl phase run initinfra-target` deploys Metal3, Calico, and Helm-Operator
3. `airshipctl baremetal clusterctl-init-target` deploys CAPI controllers
4. `airshipctl phase run clusterctl-move` pivot target cluster to CAPI self-management



Step 2:

Provision Target Cluster

1. `airshipctl phase run controlplane-ephemeral` provisions the target control plane
2. `airshipctl phase run initinfra-target` deploys Metal3, Calico, and Helm-Operator
3. `airshipctl baremetal clusterctl-init-target` deploys CAPI controllers
4. `airshipctl phase run clusterctl-move` pivot target cluster to CAPI self-management
5. `airshipctl phase run workers-target` provisions the rest of the target cluster nodes



Step 3:

Workload Deployment

1 `airshipctl phase run workload-target` **deploys software workloads to the cluster**

airship

Onboarding Materials

Get Ramped Up!

- Airship documentation: https://docs.airshipit.org
- Airship Blog Series: https://airshipit.org/blog
- Airshipctl project and GitHub issues
- Treasuremap project and GitHub issues
- AirshipUI project and GitHub issues

- Kustomize Airship Plugins video from the Summit
- Airship 101 video from the Open Infra Summit

- Layering and de-duplication doc (WIP)
- Airship-in-a-Pod changeset (WIP)

Get Involved!

- Airship Wiki
- airship-discuss@lists.openstack.org
- Freenode IRC: #airshipit
- Airshipit.org/slack: #airshipit

- IRC/Slack team meetings are every other Tuesdays
- Design meetings are Tuesdays & Thursdays
- Scope grooming meetings are Wednesdays

Community Channels

Mailing Lists: lists.airshipit.org

Freenode IRC: #airshipit

Airshipit.org/slack: #airshipit

Website: www.airshipit.org

Wiki: https://wiki.openstack.org/wiki/Airship

Documentation: https://docs.airshipit.org

OpenDev: https://opendev.org/airship

YouTube: https://www.youtube.com/user/OpenStackFoundation/