Adaptive Memory Scaling for Robust Airflow Pipelines

Cyrus Dukart, David Sacerdote &
Jason Bridgemohansingh

What we are talking about today

1. Who we are
2. Our Data Challenge
3. Our solution

vibrant planet


Who we are?

vibrant planet

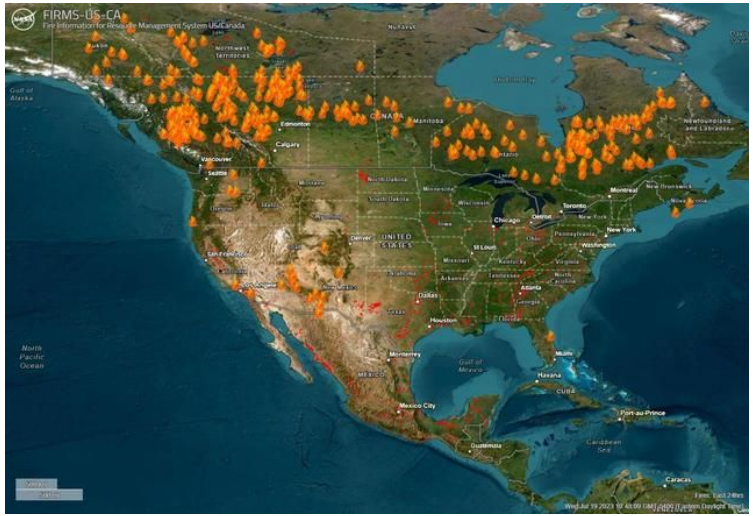
Vibrant Planet

The first common operating picture for wildfire & ecosystem resilience

The New York Times

Climate Scientists Warn of a 'Global Wildfire Crisis'

Worsening heat and dryness could lead to a 50 percent rise in off-the-charts fires, according to a United Nations report.



The Washington Post
Democracy Dies in Darkness

Massive wildfires are a new threat to Chile. Here's why they're so deadly.

By Scott Dance

Updated February 5, 2024 at 7:56 p.m. EST | Published February 5, 2024 at 4:35 p.m. EST

Increasingly hotter fires

Catastrophic effects to soils and streams

Caldor Fire:
Lake Tahoe,
2021

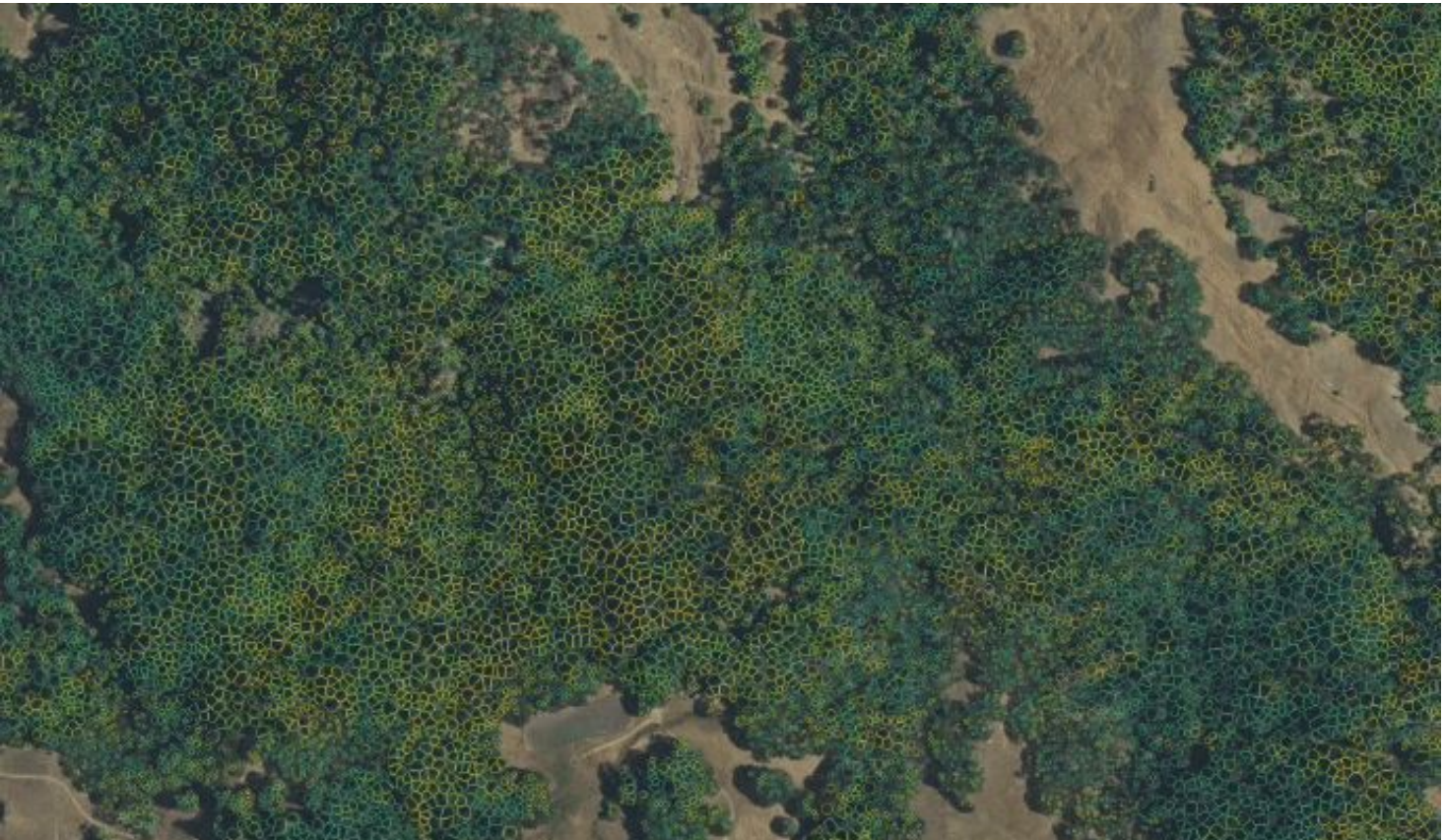
vibrant planet

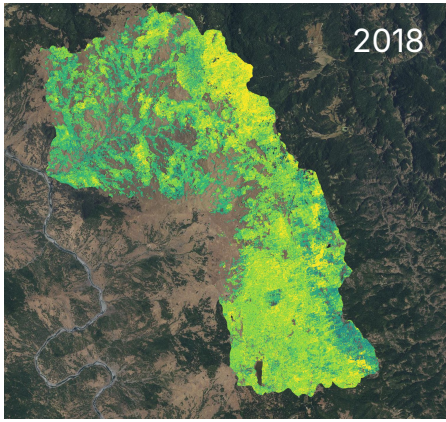
Individual tree health to forest resilience

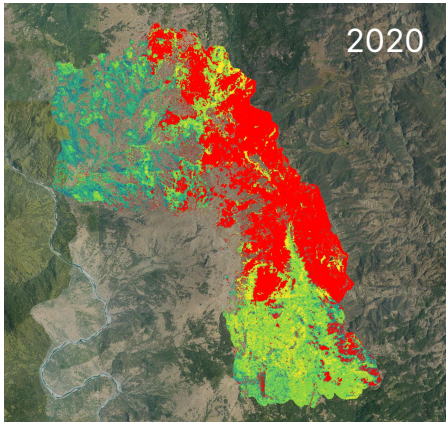
Using Synthetic Canopy Height Models to Segment individual Trees

Individual tree health to forest resilience

Using Synthetic Canopy Height Models to Segment individual Trees



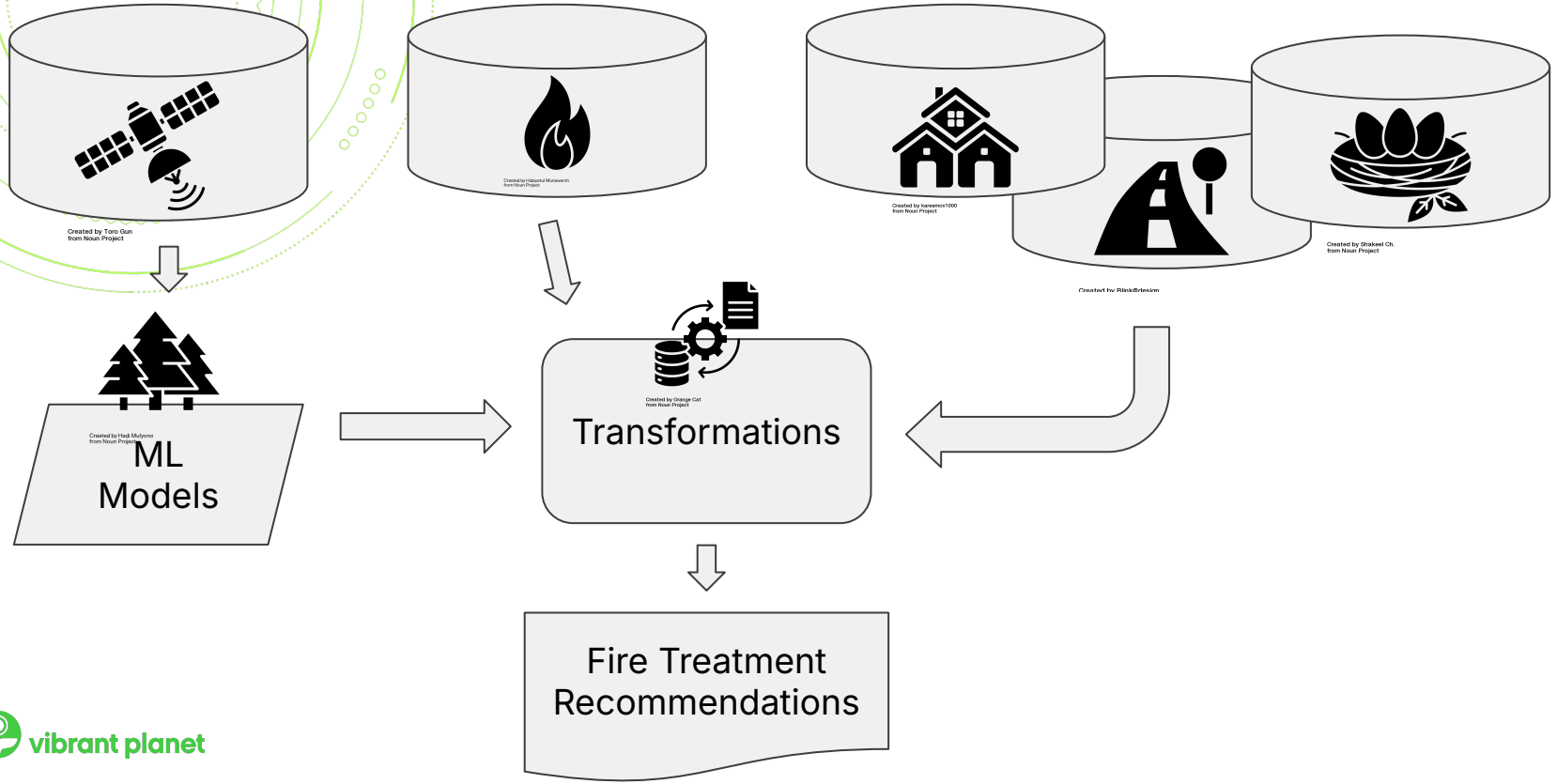




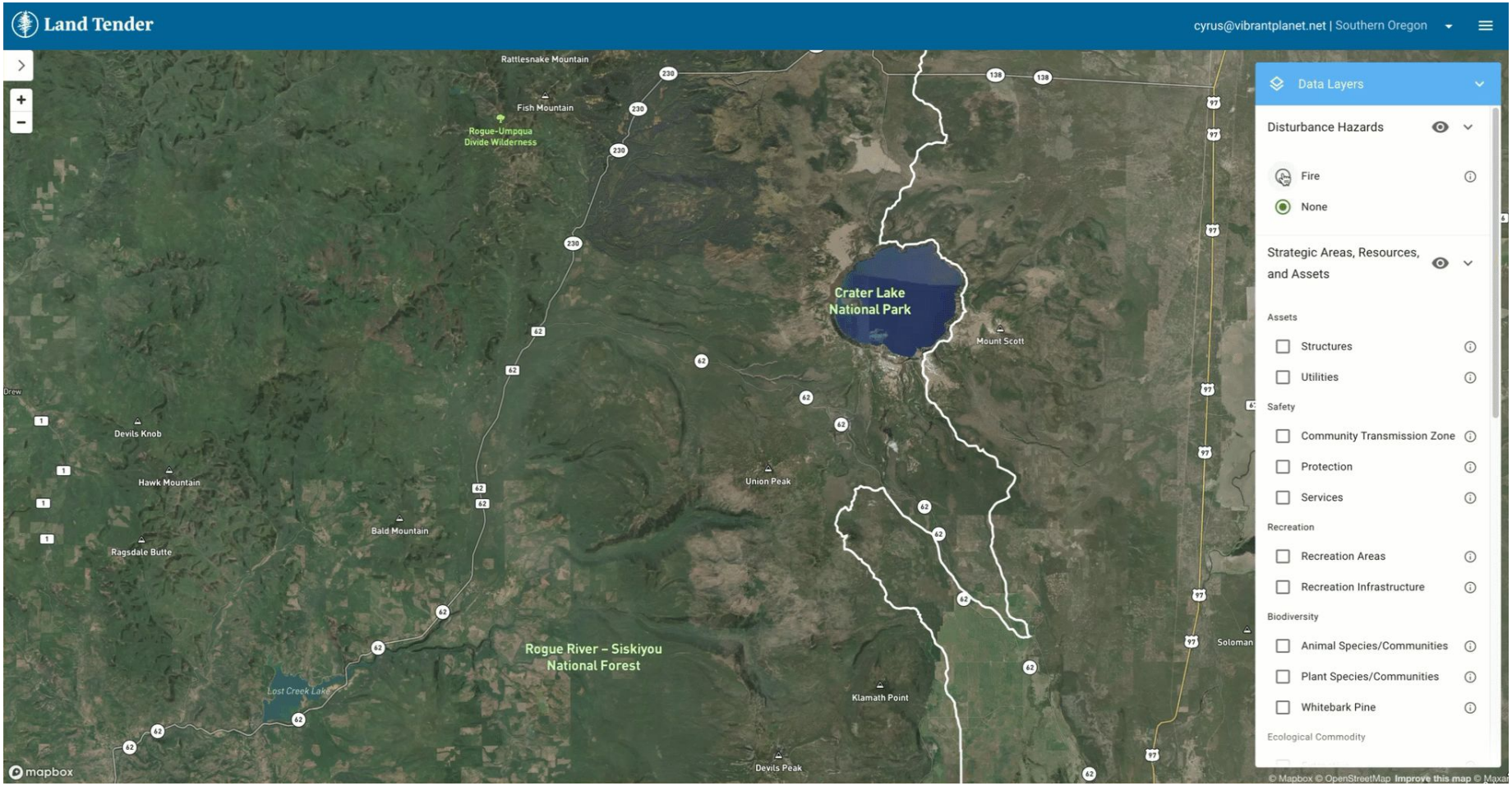
The answer: Speeding and scaling what we know works

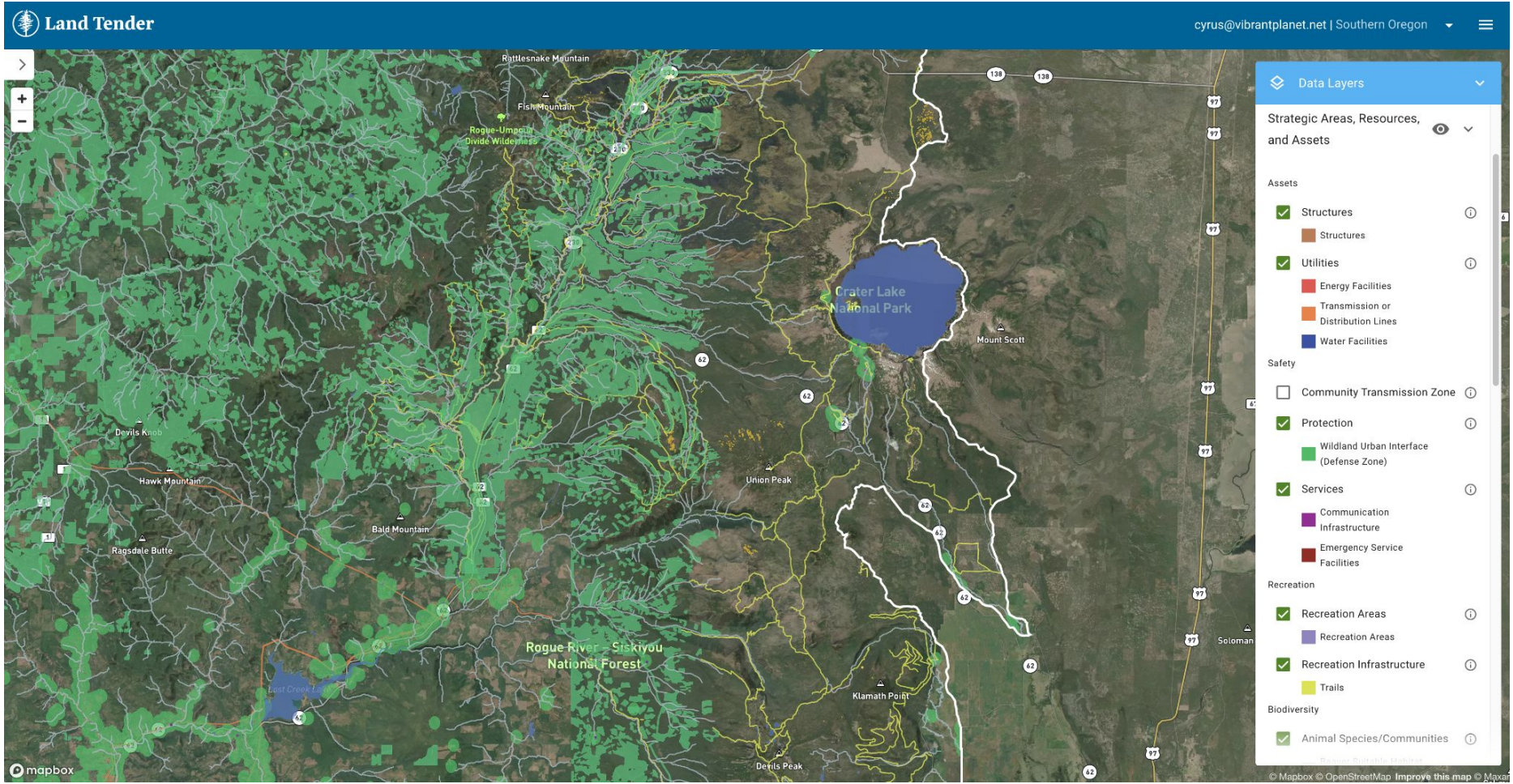
Credit: Steve Rondeau, Natural Resources Director of the Klamath Tribes

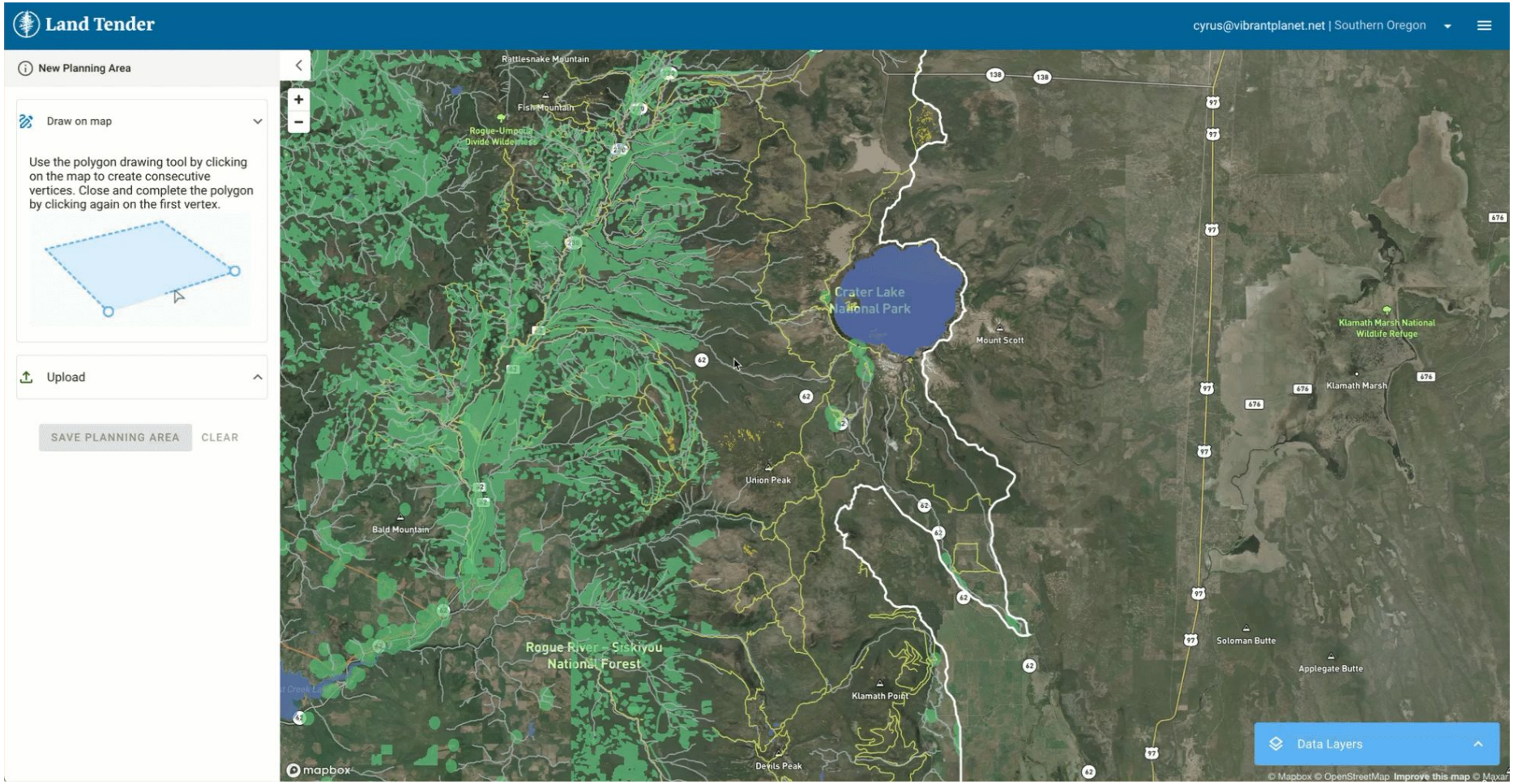
What we use Airflow for

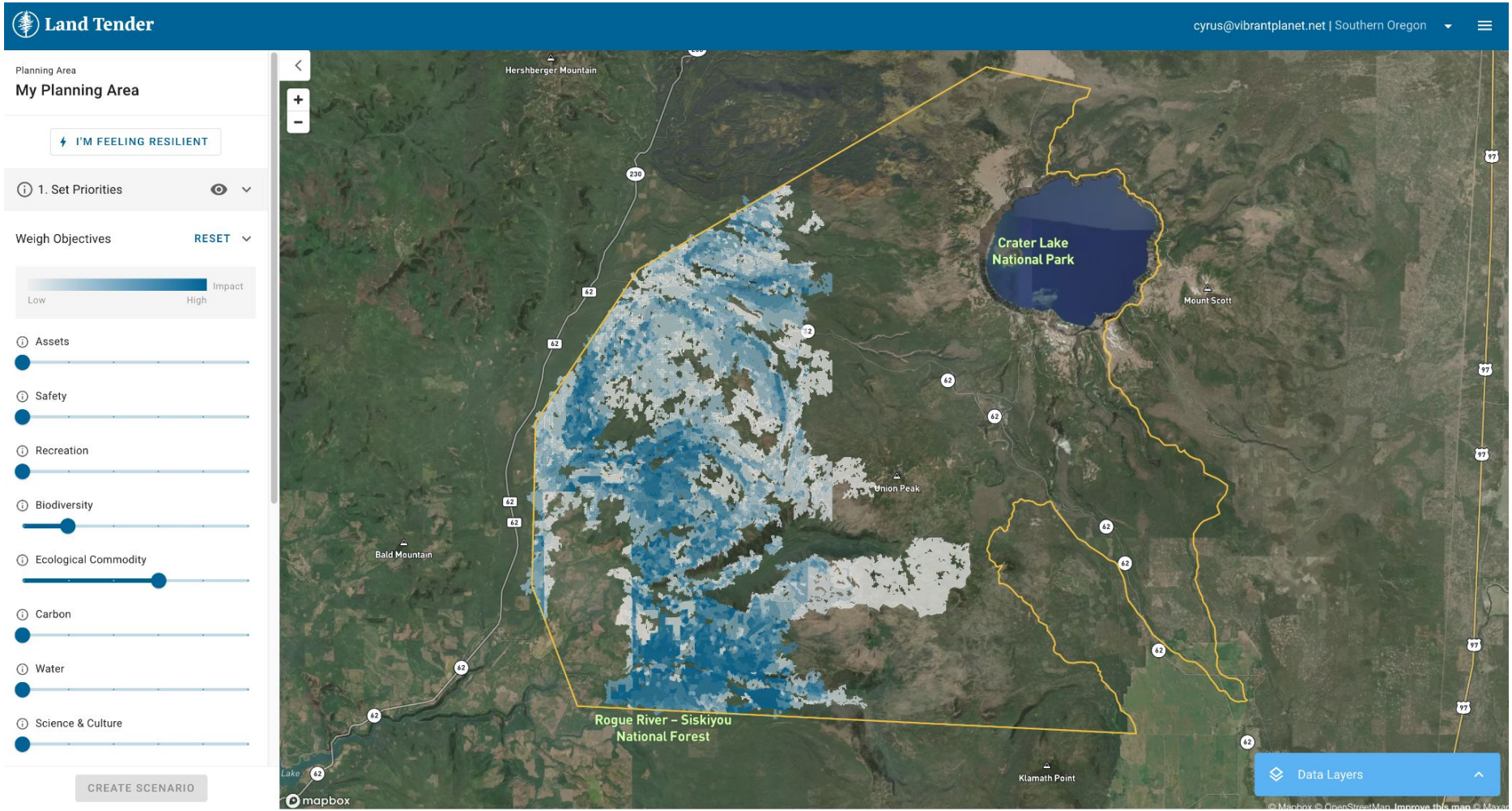


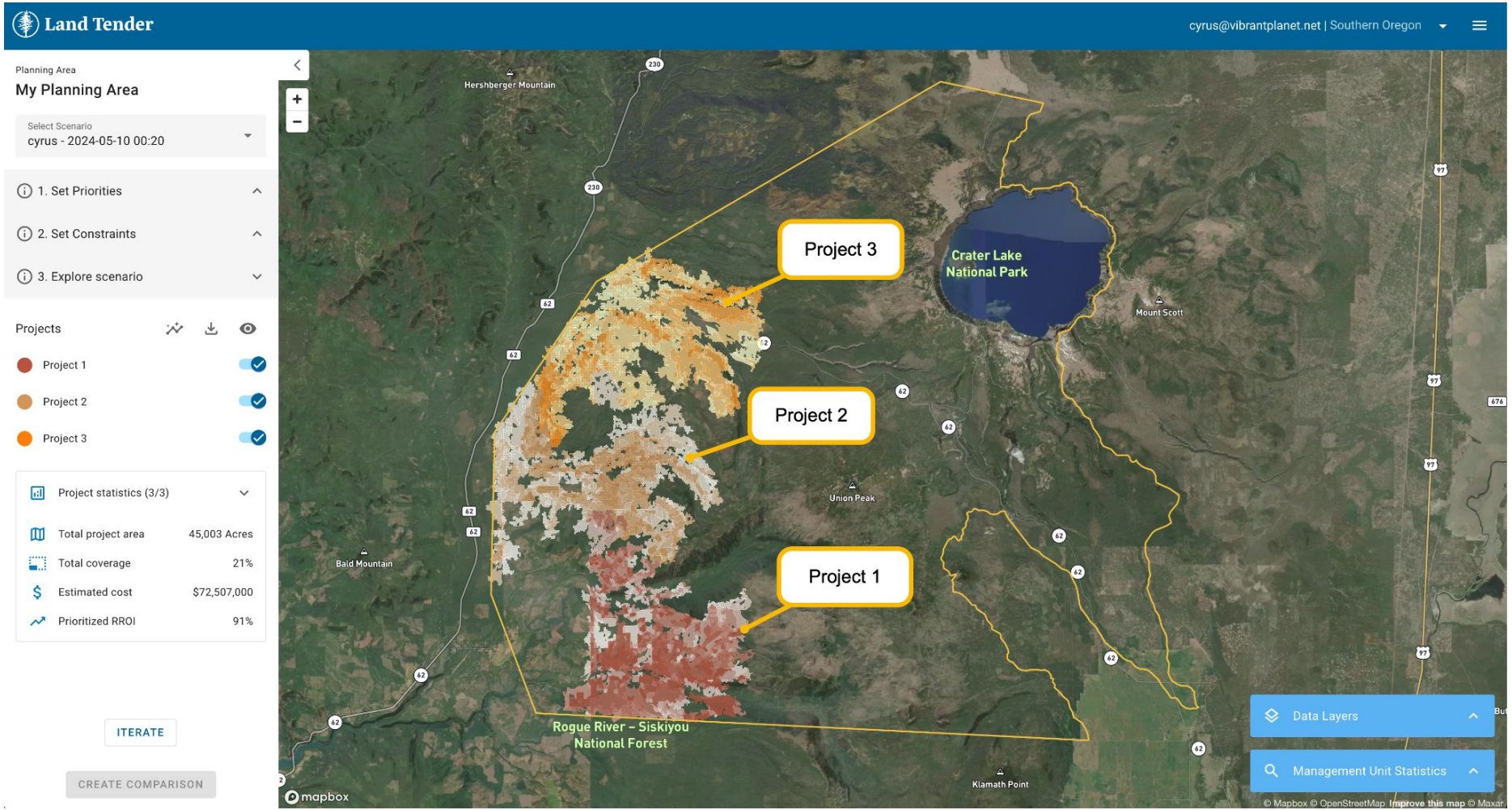
vibrant planet

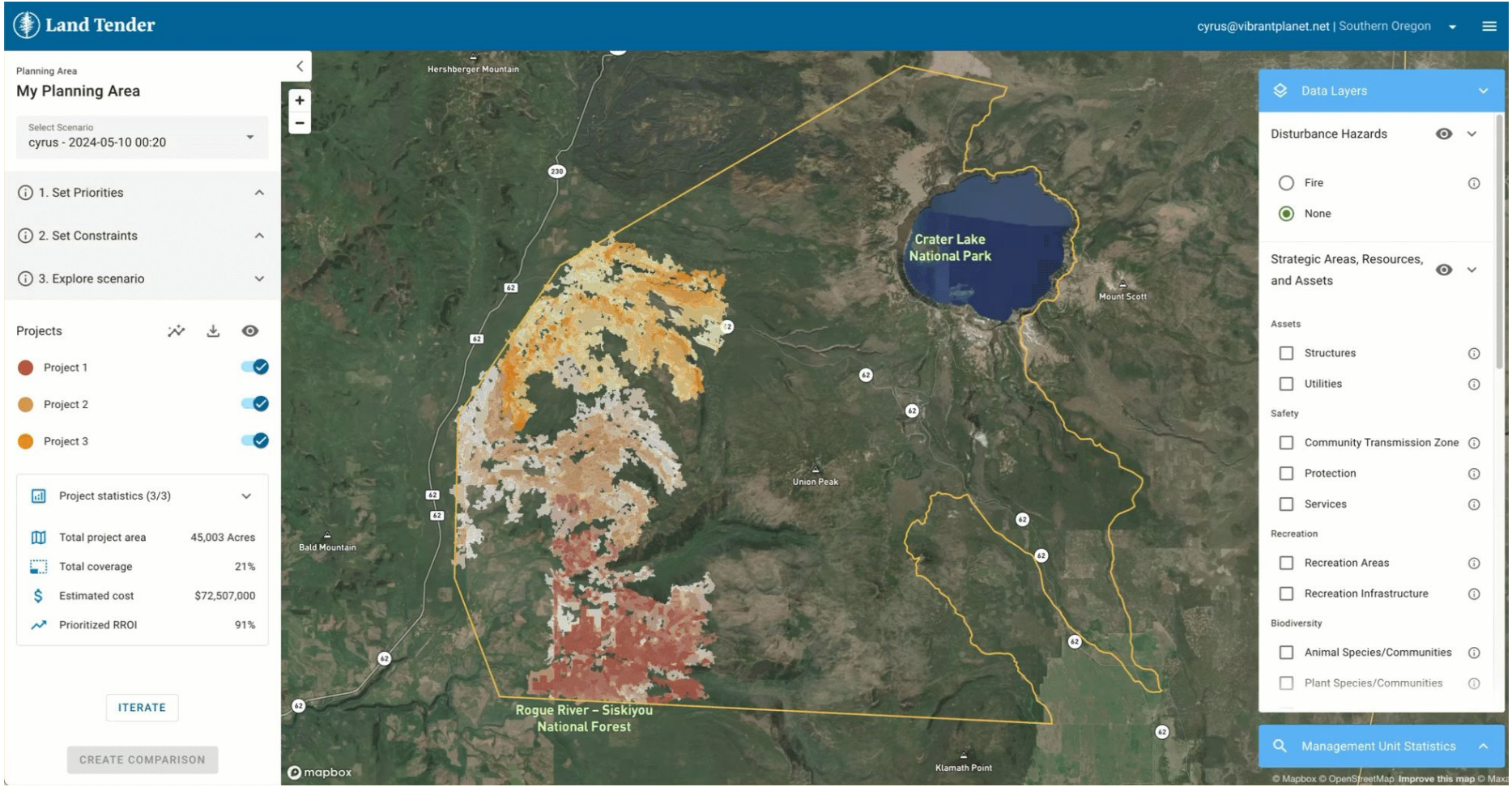

Land Tender
cyrus@vibrantplanet.net | Southern Oregon
Rattlesnake Mountain
Fish Mountain
Rogue-Umpqua Divide Wilderness
Crater Lake National Park
Mount Scott
Devils Knob
Hawk Mountain
Ragsdale Butte
Bald Mountain
Union Peak
Rogue River – Siskiyou National Forest
Lost Creek Lake
Klamath Point
Devils Peak
Soloman
Data Layers
Disturbance Hazards
Fire
None
Strategic Areas, Resources, and Assets
Assets
Structures
Utilities
Safety
Community Transmission Zone
Protection
Services
Recreation
Recreation Areas
Recreation Infrastructure
Biodiversity
Animal Species/Communities
Plant Species/Communities
Whitebark Pine
Ecological Commodity
mapbox
© Mapbox © OpenStreetMap Improve this map © Maxar


Land Tender
cyrus@vibrantplanet.net | Southern Oregon
Rattlesnake Mountain
Fish Mountain
Rogue-Umpqua Divide Wilderness
Crater Lake National Park
Mount Scott
Devils Knob
Hawk Mountain
Bald Mountain
Ragsdale Butte
Union Peak
Rogue River - Siskiyou National Forest
Klamath Point
Devils Peak
Soloman
Data Layers
Strategic Areas, Resources, and Assets
Assets
Structures
Structures
Utilities
Energy Facilities
Transmission or Distribution Lines
Water Facilities
Safety
Community Transmission Zone
Protection
Wildland Urban Interface (Defense Zone)
Services
Communication Infrastructure
Emergency Service Facilities
Recreation
Recreation Areas
Recreation Areas
Recreation Infrastructure
Trails
Biodiversity
Animal Species/Communities
mapbox
© Mapbox © OpenStreetMap Improve this map © Maxar


Land Tender
cyrus@vibrantplanet.net | Southern Oregon
New Planning Area
Draw on map
Use the polygon drawing tool by clicking on the map to create consecutive vertices. Close and complete the polygon by clicking again on the first vertex.
Upload
SAVE PLANNING AREA
CLEAR
Rattlesnake Mountain
Fish Mountain
Rogue-Umpqua Divide Wilderness
Crater Lake National Park
Mount Scott
Klamath Marsh National Wildlife Refuge
Klamath Marsh
Union Peak
Bald Mountain
Rogue River - Siskiyou National Forest
Klamath Point
Devils Peak
Soloman Butte
Applegate Butte
Data Layers
mapbox
© Mapbox © OpenStreetMap Improve this map © Maxar


Land Tender
cyrus@vibrantplanet.net | Southern Oregon
Planning Area
My Planning Area
I'M FEELING RESILIENT
1. Set Priorities
Weigh Objectives
RESET
Impact
Low
High
Assets
Safety
Recreation
Biodiversity
Ecological Commodity
Carbon
Water
Science & Culture
CREATE SCENARIO
Hershberger Mountain
Crater Lake National Park
Mount Scott
Union Peak
Bald Mountain
Rogue River – Siskiyou National Forest
Klamath Point
Data Layers
mapbox


Land Tender
cyrus@vibrantplanet.net | Southern Oregon
Planning Area
My Planning Area
Select Scenario
cyrus - 2024-05-10 00:20
1. Set Priorities
2. Set Constraints
3. Explore scenario
Projects
Project 1
Project 2
Project 3
Project statistics (3/3)
Total project area 45,003 Acres
Total coverage 21%
Estimated cost $72,507,000
Prioritized RROI 91%
ITERATE
CREATE COMPARISON
Project 3
Project 2
Project 1
Crater Lake National Park
Rogue River – Siskiyou National Forest
Hershberger Mountain
Mount Scott
Union Peak
Bald Mountain
Klamath Point
Data Layers
Management Unit Statistics
mapbox
© Mapbox © OpenStreetMap Improve this map © Maxar

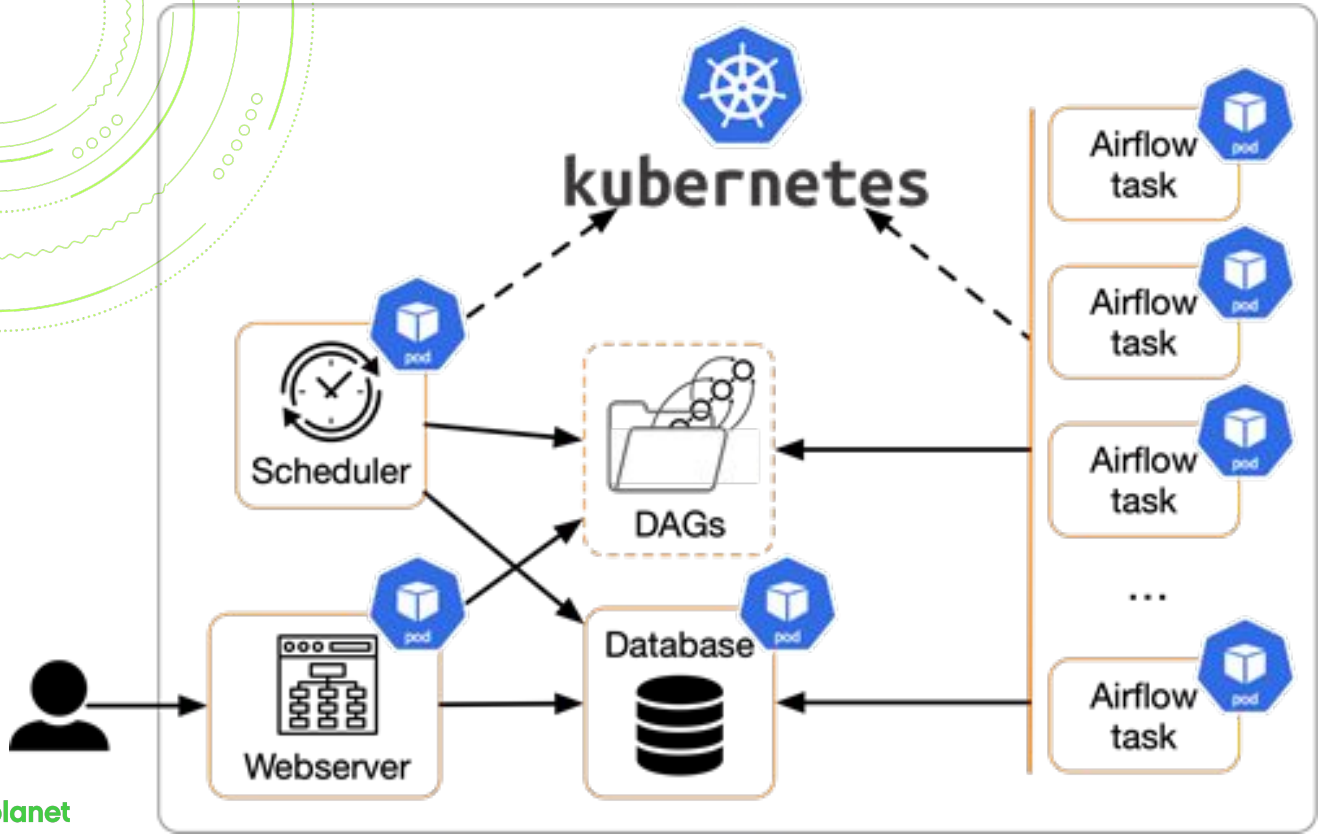

Land Tender
cyrus@vibrantplanet.net | Southern Oregon
Planning Area
My Planning Area
Select Scenario
cyrus - 2024-05-10 00:20
1. Set Priorities
2. Set Constraints
3. Explore scenario
Projects
Project 1
Project 2
Project 3
Project statistics (3/3)
Total project area 45,003 Acres
Total coverage 21%
Estimated cost $72,507,000
Prioritized RROI 91%
ITERATE
CREATE COMPARISON
Hershberger Mountain
Crater Lake National Park
Mount Scott
Union Peak
Bald Mountain
Rogue River – Siskiyou National Forest
Klamath Point
Data Layers
Disturbance Hazards
Fire
None
Strategic Areas, Resources, and Assets
Assets
Structures
Utilities
Safety
Community Transmission Zone
Protection
Services
Recreation
Recreation Areas
Recreation Infrastructure
Biodiversity
Animal Species/Communities
Plant Species/Communities
Management Unit Statistics
mapbox
© Mapbox © OpenStreetMap Improve this map


Our Data Challenge?

Airflow Executor Options

VS

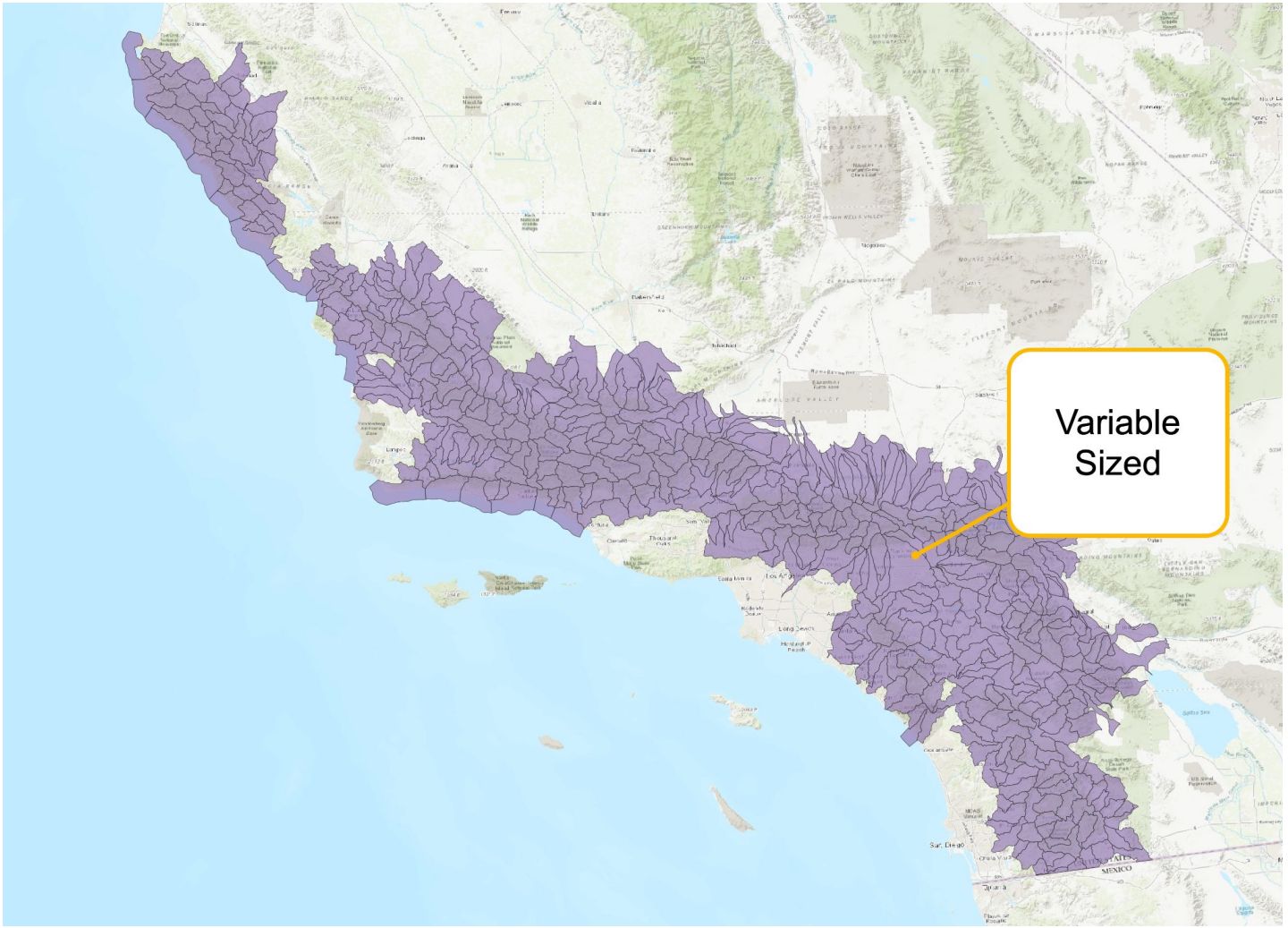
How Airflow allocates resources



vibrant planet

Our Partitioning Scheme

Variable Memory Reqs

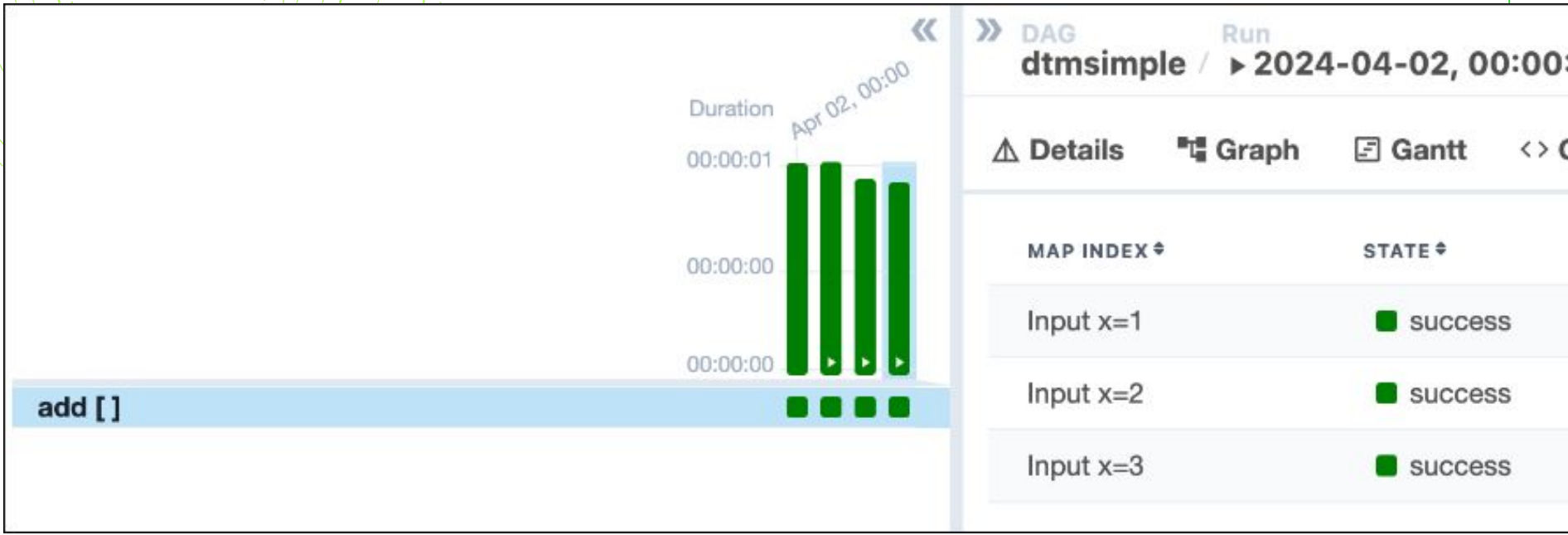


vibrant planet

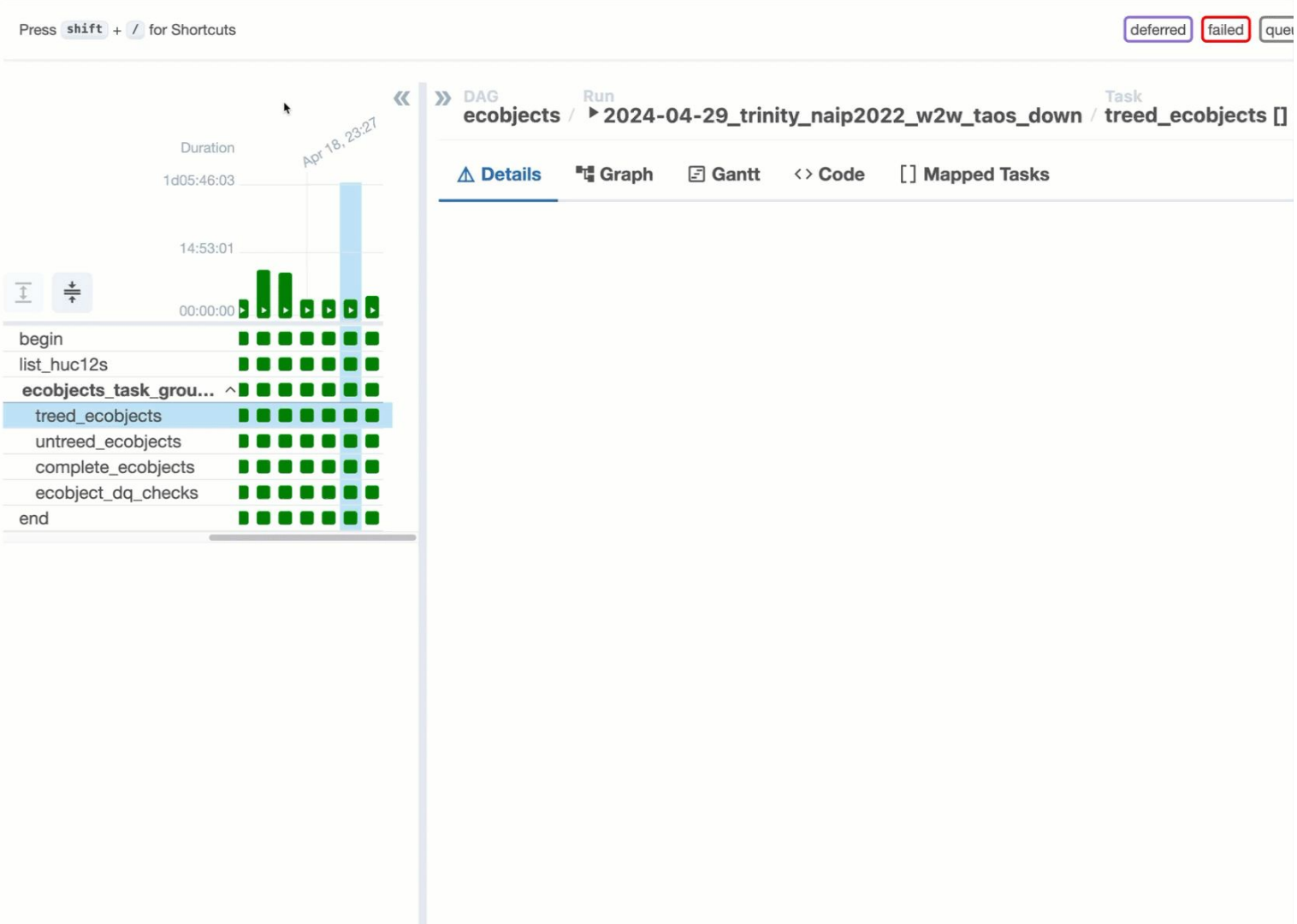
Dynamic Task Mapping

```
@task()
def add(x: int):
    logging.info(f“Running for {x}”)

expansion_example = add.expand(x=[1, 2, 3])
```

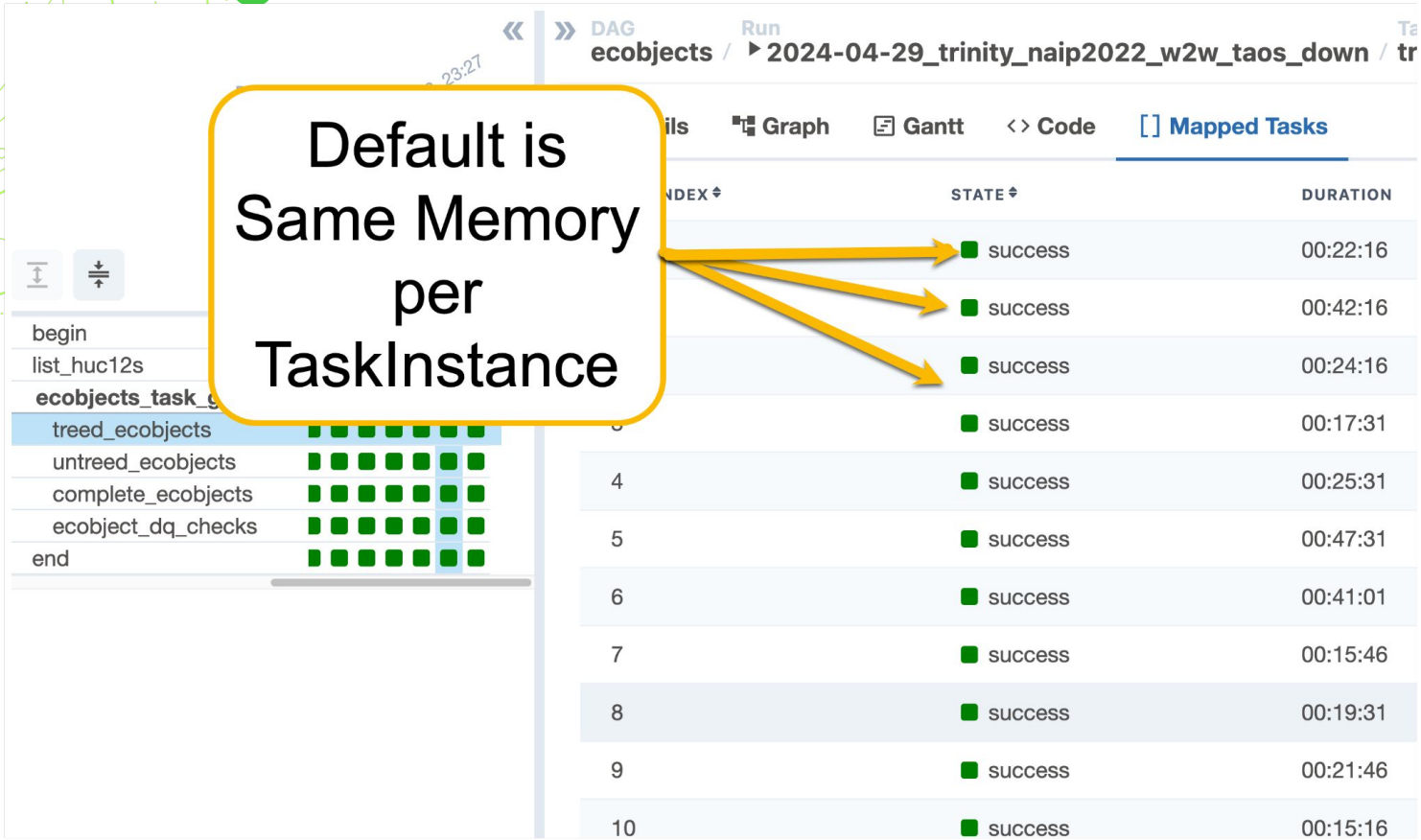
Dynamic Task Mapping



vibrant planet


Press shift + / for Shortcuts
deferred
failed
Duration
Apr 18, 23:27
1d05:46:03
14:53:01
00:00:00
begin
list_huc12s
ecobjects_task_grou...
treed_ecobjects
untreed_ecobjects
complete_ecobjects
ecobject_dq_checks
end
DAG
ecobjects
Run
2024-04-29_trinity_naip2022_w2w_taos_down
Task
treed_ecobjects []
Details
Graph
Gantt
Code
Mapped Tasks

The Challenge



Default is Same Memory per TaskInstance

vibrant planet

What's in a k8s pod spec

- Contains a full description of the environment and resource limits

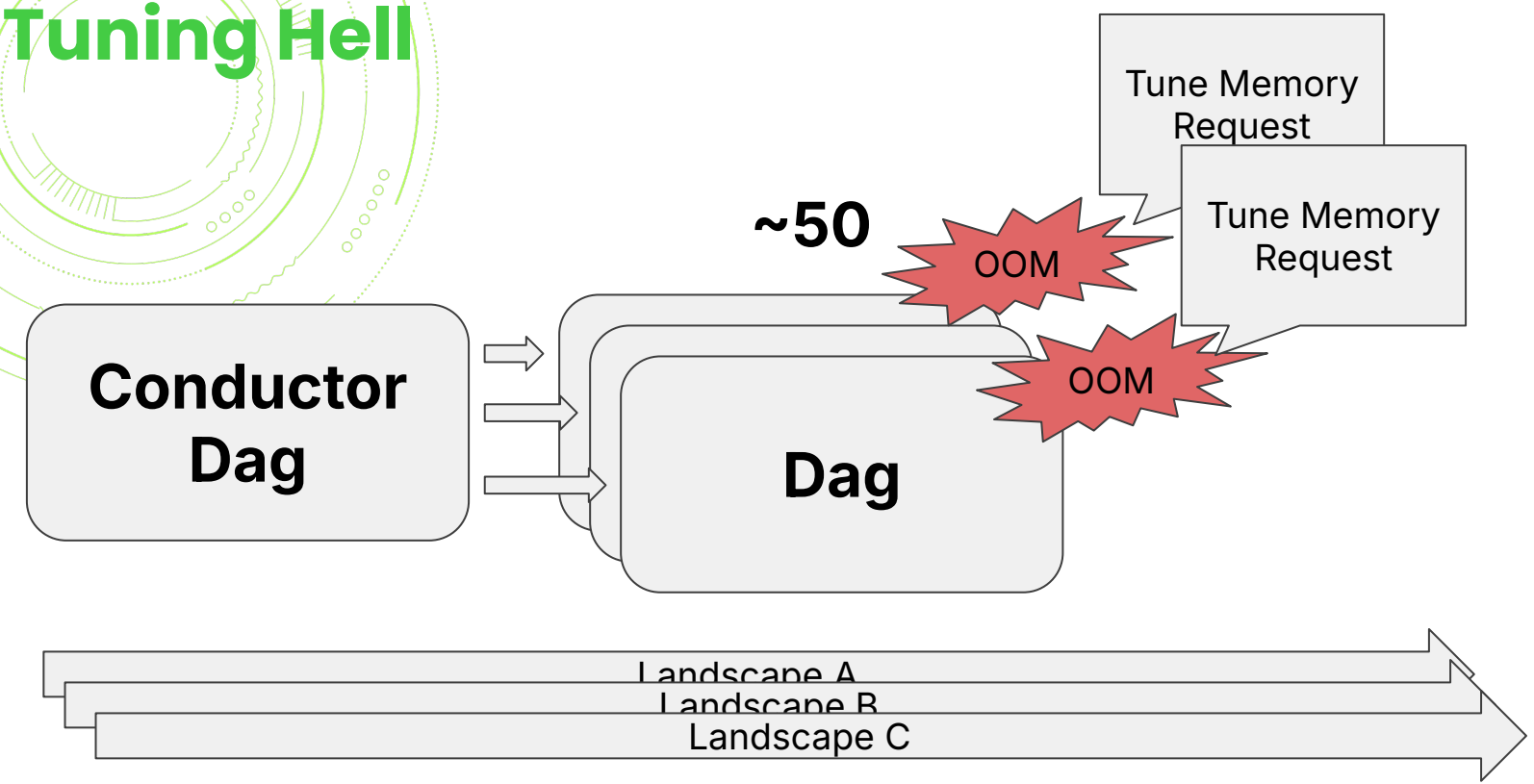
```
resources:
    limits:
      cpu: '12'
      memory: 24Gi
    requests:
      cpu: '12'
      memory: 24Gi
```

Example Task w Executor Config Override

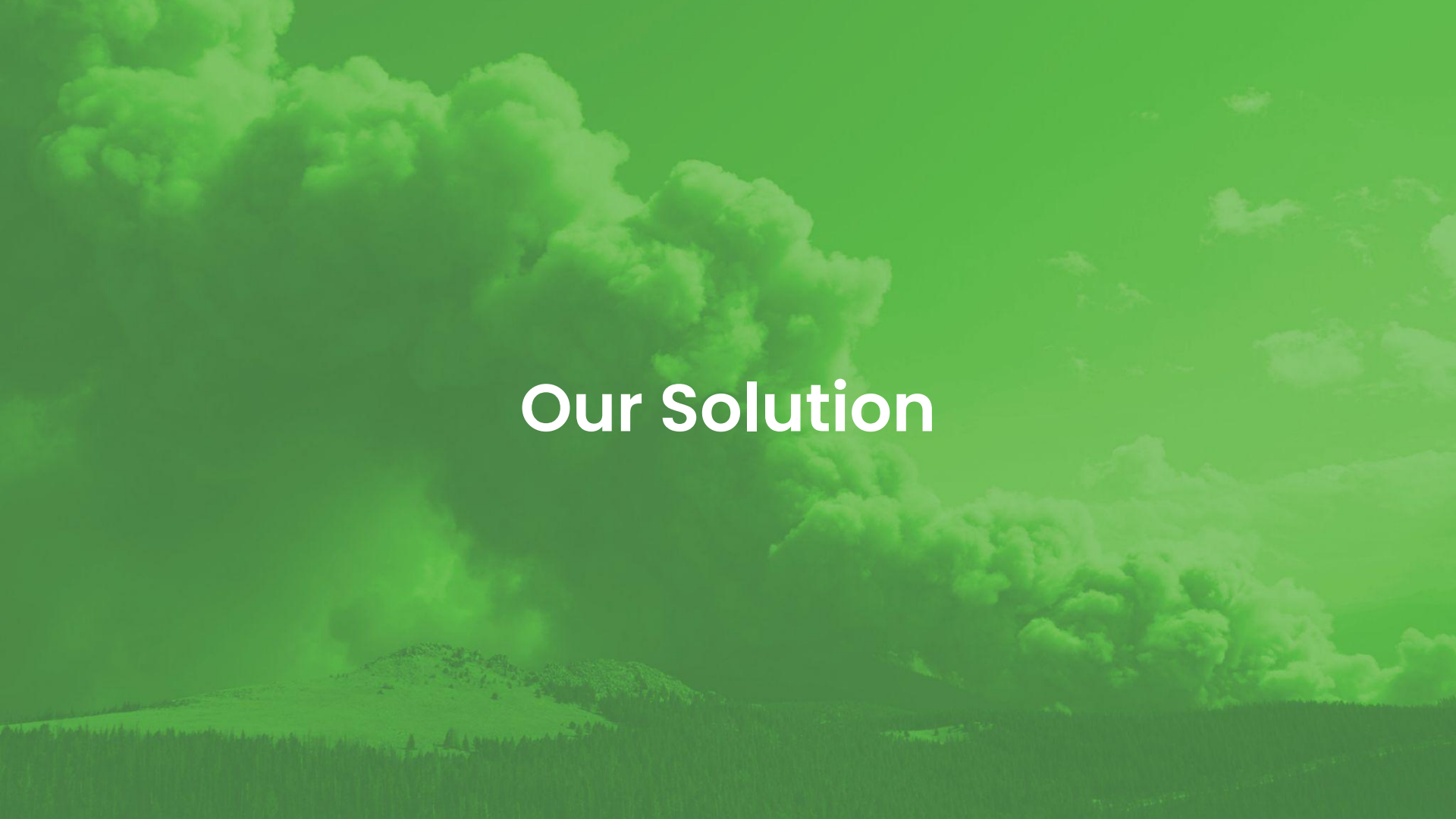
```
from kubernetes.client import models as k8s

@task(
    executor_config={
        "pod_override": k8s.V1Pod(...)
    }
)
def do_something():
        print(“my resources were customized!”)
```

vibrant planet

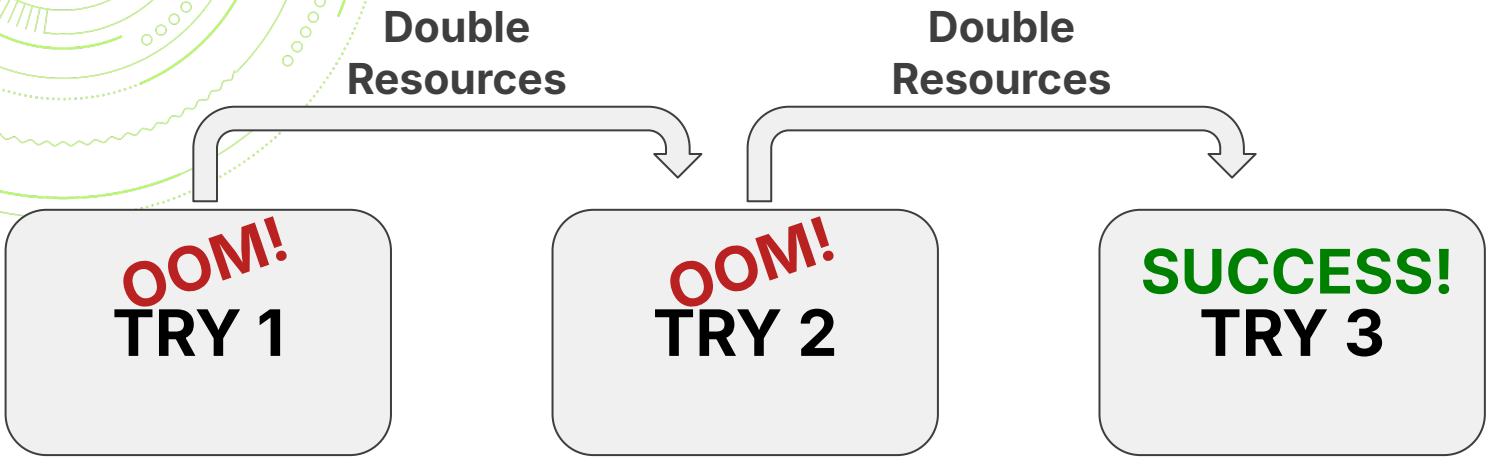
Tuning Hell



vibrant planet


Our Solution

Make Self Healing Pipelines



Solution: Mutate Pod Executor Config

```
from kubernetes.client import models as k8s

def update_task_instance_pod_override(
    ti: TaskInstance,
    executor_config: Dict[str, V1Pod]
) -> None:
    with create_session() as session:
        ti.executor_config = executor_config
        session.add(ti)
    ti.refresh_from_db()
```

vibrant planet

Solution: Example Executor Config

```
executor_config ={
    "pod_override": k8s.V1Pod(
        spec=k8s.V1PodSpec(
            containers=[
                k8s.V1Container(
                    name="base",
                    resources=k8s.V1ResourceRequirements(
                        requests=K8sResources(cpu="4", memory="8Gi"),
                        limits=K8sResources(cpu="4", memory="8Gi"),
                    ),
                )
            ],
        ),
    )
}
```

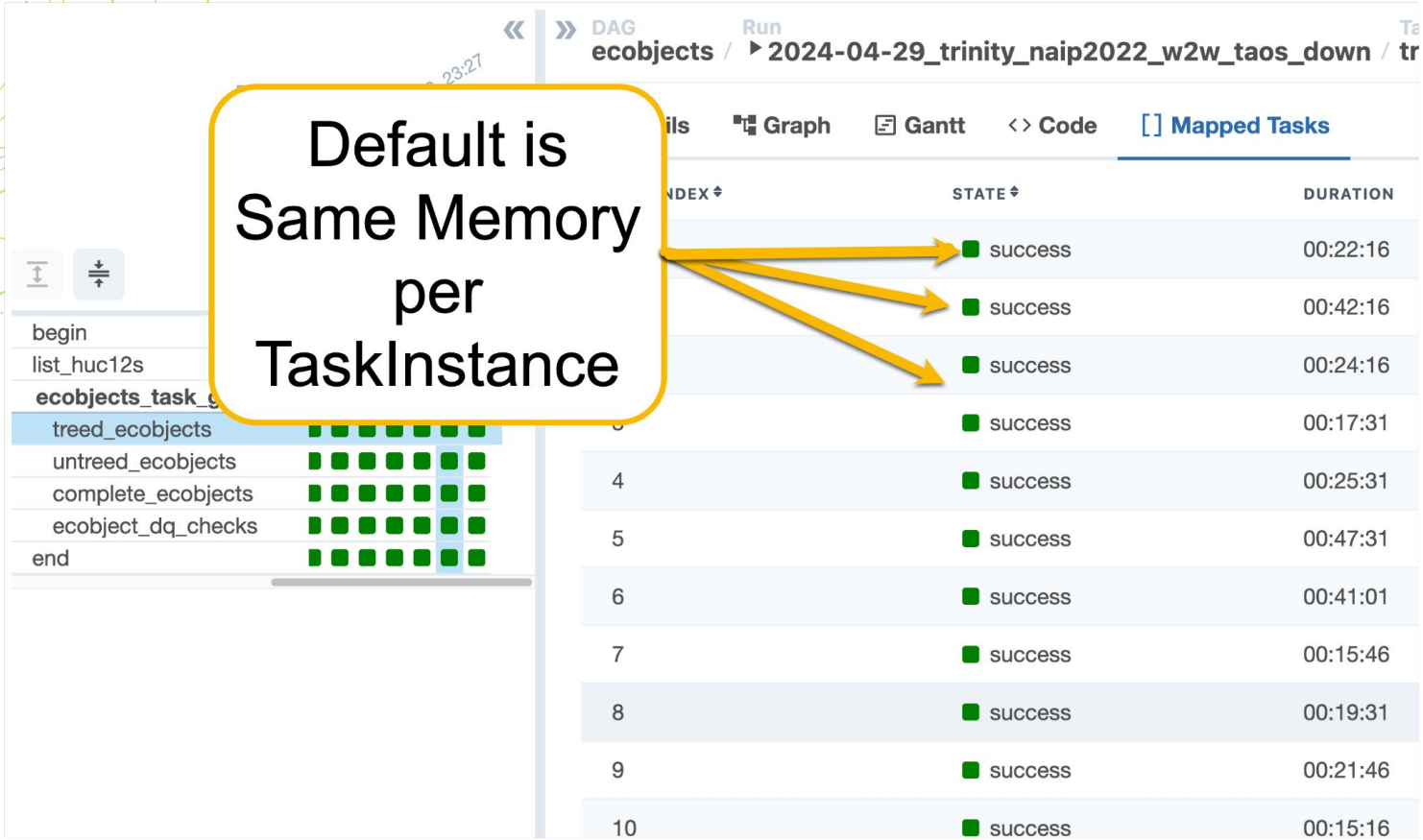
Solution: Example Dag

```
@dag()
def example_dag():

    @task(
        retries=3,
        on_retry_callback=double_memory()
    )
    def failing_task():
        raise ValueError("Failing to Memory Ramp")
```


But wait there’s more

What about initial Memory Allocation

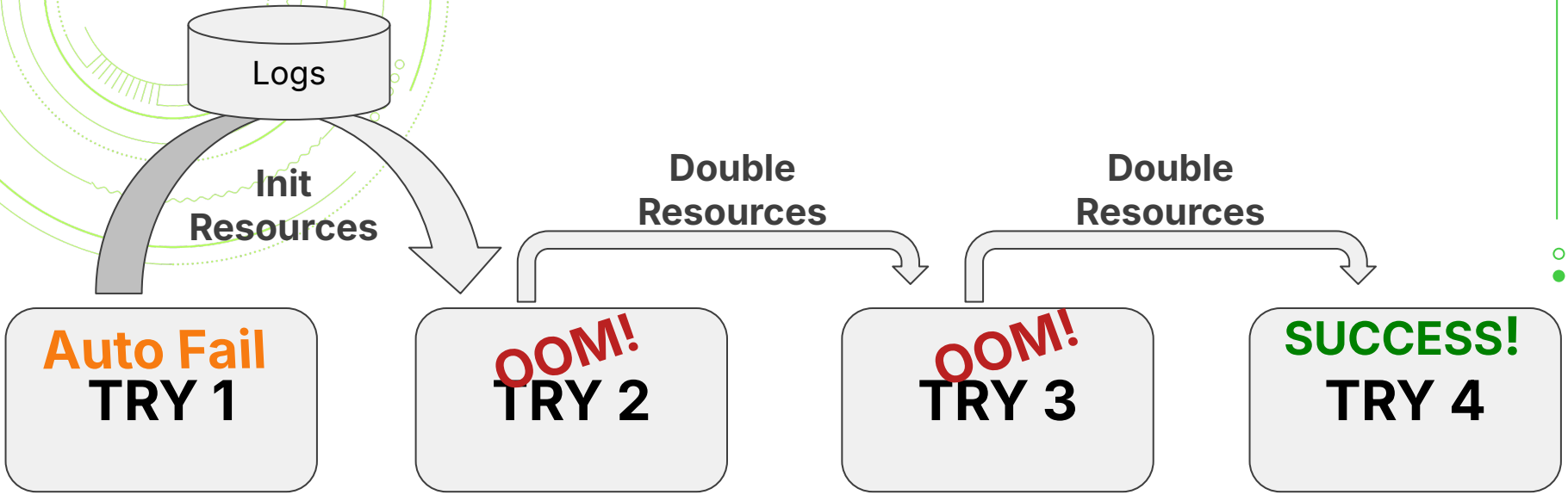
The Problem



Default is Same Memory per TaskInstance

vibrant planet

Set Initial Resources per TI



vibrant planet

Implementation: first try fail

Because we can't set memory allocation before we run, the first attempt to run any task is automatically failed as a means of setting the correct amount of memory

```
[2024-02-23, 23:41:01 UTC] {vp_python.py:202} INFO - Running with the following memory settings:
[2024-02-23, 23:41:01 UTC] {vp_python.py:203} INFO -     Total memory from Executor Config: 9.0 GiB
[2024-02-23, 23:41:01 UTC] {vp_python.py:206} INFO -     Total memory available: 9.0 GiB
[2024-02-23, 23:41:01 UTC] {vp_python.py:209} INFO -     Total memory limit on process: 8.1 GiB
[2024-02-23, 23:41:01 UTC] {airflow_helpers.py:200} INFO - Recommending default memory for huc12_id =
[2024-02-23, 23:41:01 UTC] {airflow_helpers.py:435} INFO - Setting Memory for Executor Config to: 4.0
GiB
[2024-02-23, 23:41:02 UTC] {airflow_helpers.py:498} INFO - ############################
[2024-02-23, 23:41:02 UTC] {airflow_helpers.py:499} INFO - # EXPECTED FIRST TRY FAIL #
[2024-02-23, 23:41:02 UTC] {airflow_helpers.py:500} INFO - ############################
```

vibrant planet

How we make an initial estimate

Task Run before?	Data (HUCs) Run Before?	Action
✅	✅	Allocate prior memory consumption + safety margin
✅	❌	Run a linear regression to estimate memory needs; add safety margin
❌	❌	Make a guess that works for a lot of tasks

Custom Decorator

We implement this using a custom decorator, @task.vp

```
@task.vp
def save_the_forest():
    from utils.forest_save import forest_saver_2000
    forest_saver_2000()
```

Custom Decorator Goodies

1. Stats Logging
2. Initial Memory Recommendation
3. Memory Ramp on Retry
4. OOM Detection

Q&A

Appendix

vibrant planet

Self Healing Pipelines
(OOM Resilience)

OOM detection and retry

OOM frequently results in a process death with no direct notification

So we wrap our tasks in a parent process which detects OOM situations and kicks off a retry with double the amount of memory

This parent process also collects memory and CPU utilization statistics during process execution and logs them to a database upon completion.

Small job optimization

Jobs with low memory needs (eg: smaller than the memory-estimation stub) get launched inside the same pod and memory footprint of the memory-estimation stub, rather than needing a failure and retry. This reduces both both the time and expense of running them.

Short-running jobs are often batched together, avoiding the overhead of launching new containers.