SOAFEE

Porting the SOAFEE AVP Blueprint to a China Cloud

A Practical Migration Journey

Wei Chen， Arm China
NOV 12, 2025

What is the SOAFEE AVP Blueprint?

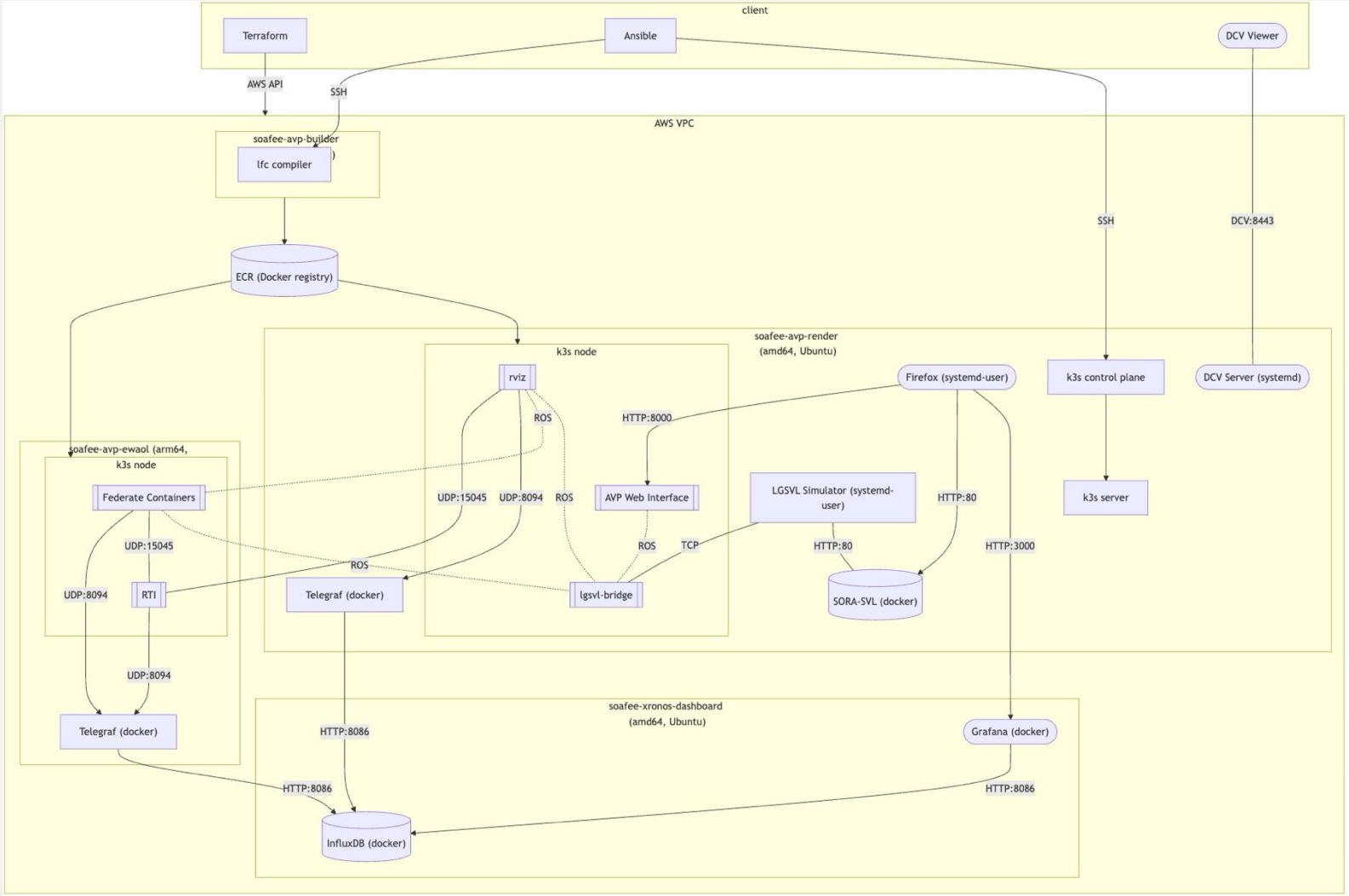
AVP Blueprint Overview

Originally developed by DENSO as an open reference blueprint for the SOAFEE community.

- Hosted at:
  - https://github.com/DensoSVIC/soafee-avp-blueprint
- Demonstrates a complete Autonomous Valet Parking (AVP) scenario built on the SOAFEE architecture.
- Implements cloud-native principles for automotive workloads:
  - Modular containerized federates
  - Real-time Linux and mixed-critical orchestration
  - Edge-cloud collaboration through k3s orchestration
- Designed to run on both arm64 and x86_64 nodes
  - Showcasing SOAFEE's cross-architecture flexibility.
- Serves as the reference for deploying, testing, and extending automotive workloads using SOAFEE.

AVP Blueprint Architecture

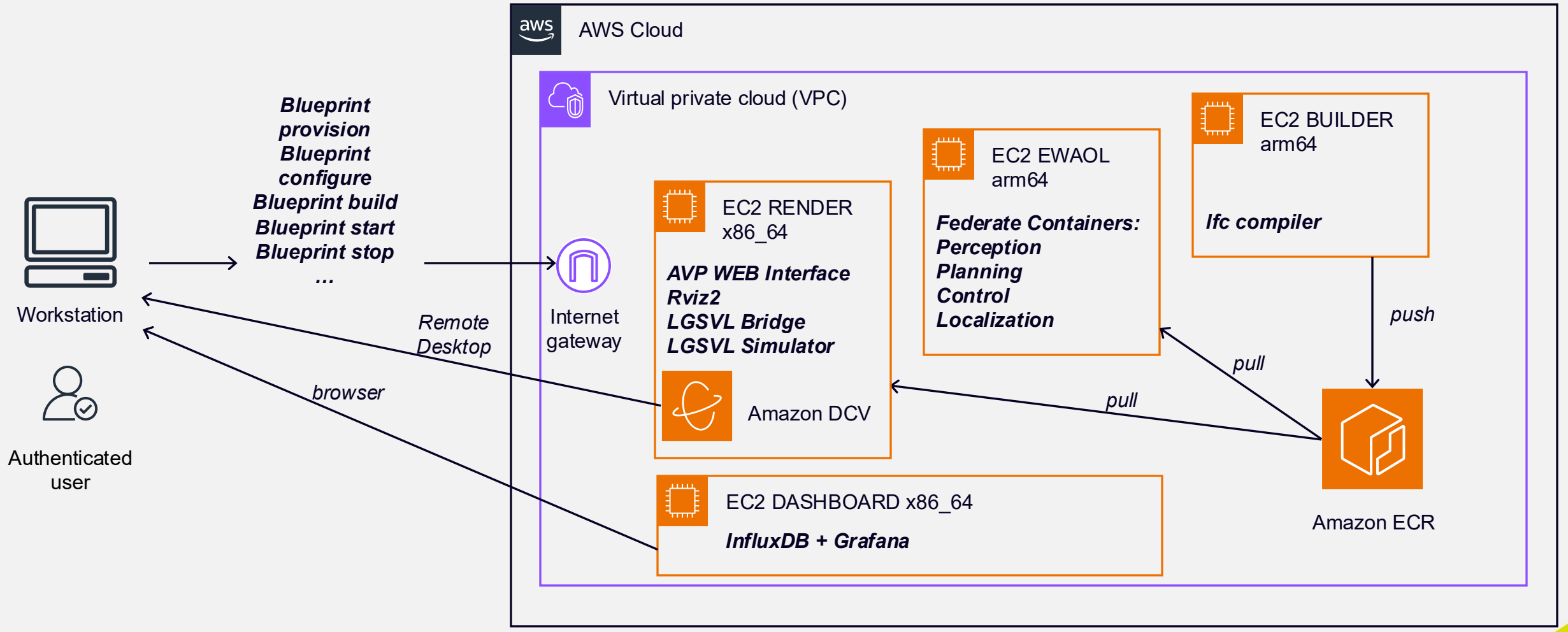
AVP Blueprint Overview



- IaC & Provisioning:
  - Terraform and Ansible.
- Image Supply:
  - Builder compiles AVP components.
  - Pushes versioned images to ECR.
- Orchestration & Simulation:
  - k3s control plane schedules AVP pods.
  - LGSVL + SORA-SVL run parking scenarios.
- ROS & Application Interfaces:
  - Bridge + rviz + AVP Web UI.
- Real-Time & Heterogeneous:
  - ARM node runs federated / RTI modules.
- Observability & Remote Access:
  - Telegraf → InfluxDB → Grafana dashboards;
  - NICE DCV streams 3D & UI to developer.

Original AWS Stack Components

SOAFEE AVP Blueprint Overview



AVP Blueprint Deployment on AWS

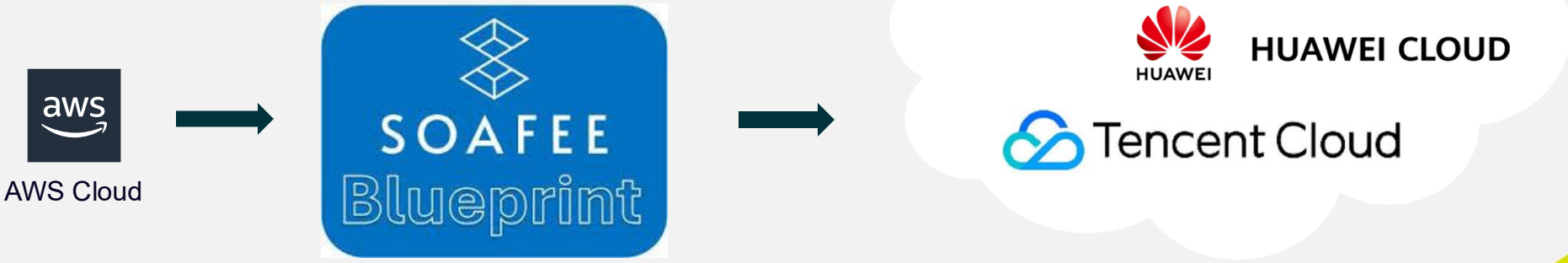
AVP Blueprint Overview

Name	Instance Type	Role	Description
EC2-RENDER	• g5.4xlarge, x86_64 • CPU: 16 • MEM: 64 GB	• SOAFEE Orchestrator • AVP Simulator • Web Interface • Telegraf agent	SOAFEE Orchestrator (k3s Master): - Manages AVP containers across nodes - Deploys workloads using k3s AVP Simulator (LGSVL) - Simulates parking environment - Publishes sensor data to federate pods Web Interface + RViz2 - User-facing GUI / Visualization of AVP simulator
EC2-EWAOL	• t4g.2xlarge, arm64 • CPU: 8 • MEM: 32 GB	• AVP workloads • Telegraf agent	AVP workloads (Federate arm64 Containers): - Perception: Lidar fusion, object detection - Localization: NDT map, robot-state-publisher - Planning: Global planner, lane planner - Control: MPC controller, behavior planner
EC2-DASHBOARD	• t3.xlarge, x86_64 • CPU: 4 • MEM: 16 GB	• Dashboard • Monitoring Services	Monitoring Services: - InfluxDB - Telegraf agents collect metrics Dashboard Services: - Grafana
EC2-BUILDER	• t4g.2xlarge, arm64 • CPU: 8 • MEM: 32 GB	• Ifc build machine	- Used to build arm64 container images during setup - Idle during runtime

Why Migrate the AVP Blueprint (AWS → China Cloud)

Drivers (Pain Points / Needs)

- Compliance & Data Sovereignty:
  - Meet local security / MLPS, cryptography, filing & audit requirements.
- Multi-Cloud Sustainability:
  - Avoid lock-in, build a decentralized, portable deployment strategy.
- Cost & Supply Chain Control:
  - Reduce reliance on overseas services.
  - Improve delivery controllability & risk mitigation.
- Technical Validation:
  - Prove SOAFEE architecture portability and behavioral consistency.
- User Experience & Latency:
  - Current AWS access from our target region suffers high RTT.



Core Objectives of the Migration

Portability, Parity, and Enhanced Developer Experience

- Minimal Code & Config Change
  - Transfer cloud-specific differences into Terraform modules & Ansible roles
  - Keep AVP application logic and scenario behavior untouched
- Performance Parity
  - Maintain startup time & perception inference latency close to the original AWS baseline
  - Keep container lifecycle stability close to the original AWS baseline
- Standardized Automation Pipeline
  - One reproducible Terraform + Ansible workflow
  - Enables one-command provisioning and prevents environment drift
- Improved Interactive User Experience
  - Lower remote visualization latency (DCV/VNC) and accelerate image pulls
  - Smooth control feedback loop for more responsive simulation & debugging

Overall Migration Strategy

- **Strategy Principles:**

Isomorphic First

Abstraction Upwards

Image Localization

Reversible Deployment

Migration Phases:

- Cloud resource selection & account compliance preparation
- Terraform provider & module adaptation
- Base image & container registry localization
- Network & security tuning
- Ansible playbook parameter & role adjustments
- AVP application layer adaptation
- Integration testing
- Demo environment deployment & graphical access verification

Core Replacement Mapping (AWS → China Cloud)

Migration Workflow

Layer / Function	AWS Service / Component	China Cloud Equivalent	Migration Action
Compute Nodes	EC2 (x86 / ARM)	ECS (x86, ARM)	Instance type mapping & sizing
GPU / Rendering	NVIDIA GRID	NVIDIA A10	VNC stack
Virtual Network	VPC + Subnets	VPC + VSwitch	Terraform module rewrite
Security Boundary	Security Groups	Security Groups	Direct mapping
Public Exposure / NAT	NAT Gateway	NAT Gateway	Module swap
Elastic IP	Elastic IP (EIP)	EIP	Direct mapping
Internal DNS	Route 53	Private Zone	Re-point zone configs
Block Storage	EBS (gp3 / io2)	ESSD (PL0–PL3)	Storage class mapping
Object Storage (if used)	S3	OSS	Adjust endpoints & credentials
Container Registry	ECR	ACR	Mirror + re-tag Push multi-arch
Identity & Access	IAM Roles / Policies	RAM Roles / Policies	Map permissions Assume-role logic
Remote GUI	NICE DCV	x11VNC	Install & optimize

Compute & Architecture Mapping

EC2 → ECS Instance Transition

	AWS				China Cloud			
Role	AWS EC2 Type	vCPU	Memory	CPU / GPU	New ECS Type	vCPU	Memory	CPU / GPU
RENDER	g5.4xlarge (x86_64)	16	64 GB	NVIDIA GRID	ecs.gn7i-c16g1.4xlarge (x86_64)	16	60 GB	NVIDIA A10
EWAOL	t4g.2xlarge (arm64)	8	32 GB	ARM64 Neoverse	ecs.g6r.2xlarge (arm64)	8	32 GB	ARM64 Neoverse
DASHBOARD	t3.xlarge (x86_64)	4	16 GB	General Purpose	ecs.g9i.xlarge (x86_64)	4	16 GB	General Purpose
BUILDER	t4g.2xlarge (arm64)	8	32 GB	ARM64 Neoverse	ecs.g6r.4xlarge (arm64)	16	64 GB	ARM64 Neoverse

Terraform Layer Adaptation

Highlights

- Provider / Credential Localization
- Networking & Core Infra
- VPC + VSwitch (public/private) recreated
  - Security Group rule model fixed
  - Instance Mapping & Normalization
- Identity and Access Management Adaptation
  - AWS AMIs → Resource and Access Management
- Container Registry
  - AWS ECR → China Cloud Container Registry
- IAM Adaptation
  - AWS IAM → China Cloud Role
- Security Group Rule Refactor

Ansible Playbooks Adjustments

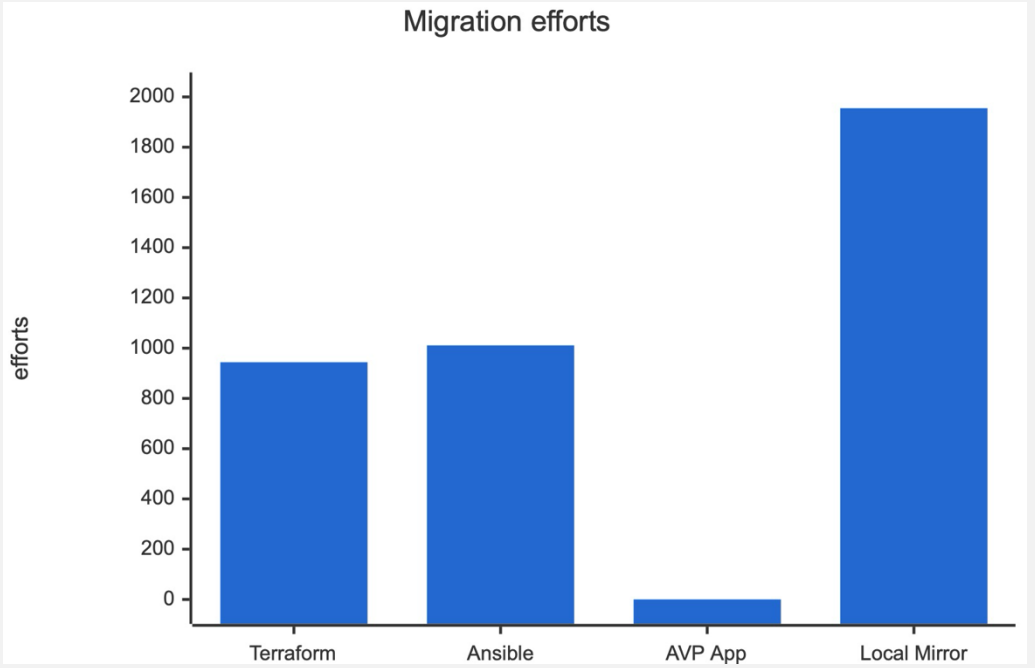
Highlights

- Cloud Toolchain & Provider Shift
  - AWS CLI → China Cloud provider CLI.
- Container Registry Authentication Adaptation
  - Add new login flow for China Cloud provider.
- GPU & Visualization Stack Adaptation
  - NVIDIA driver role parameters changed.
  - New display detection logic and VNC service.
- User / System Differences
  - Default ansible_user mappings and disk size adjustment.
- Image & Build Pipeline Adjustments
  - Container pulls switched to Local Container Registry.
- Source Assets & Demo Repository
  - AWS S3 & GITHUB → China Local Storage Provider.

Porting Challenges and Solutions

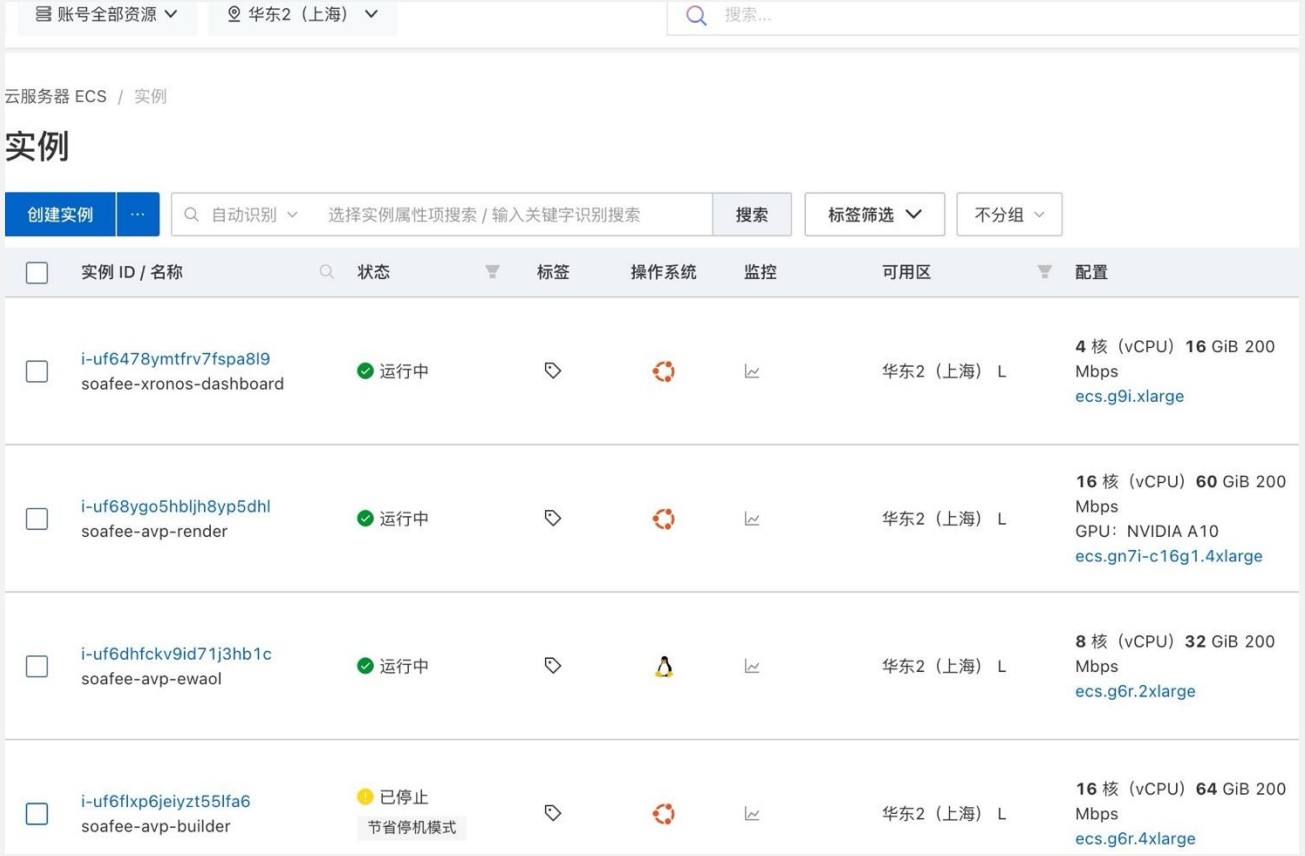
Challenges, Solutions and Efforts

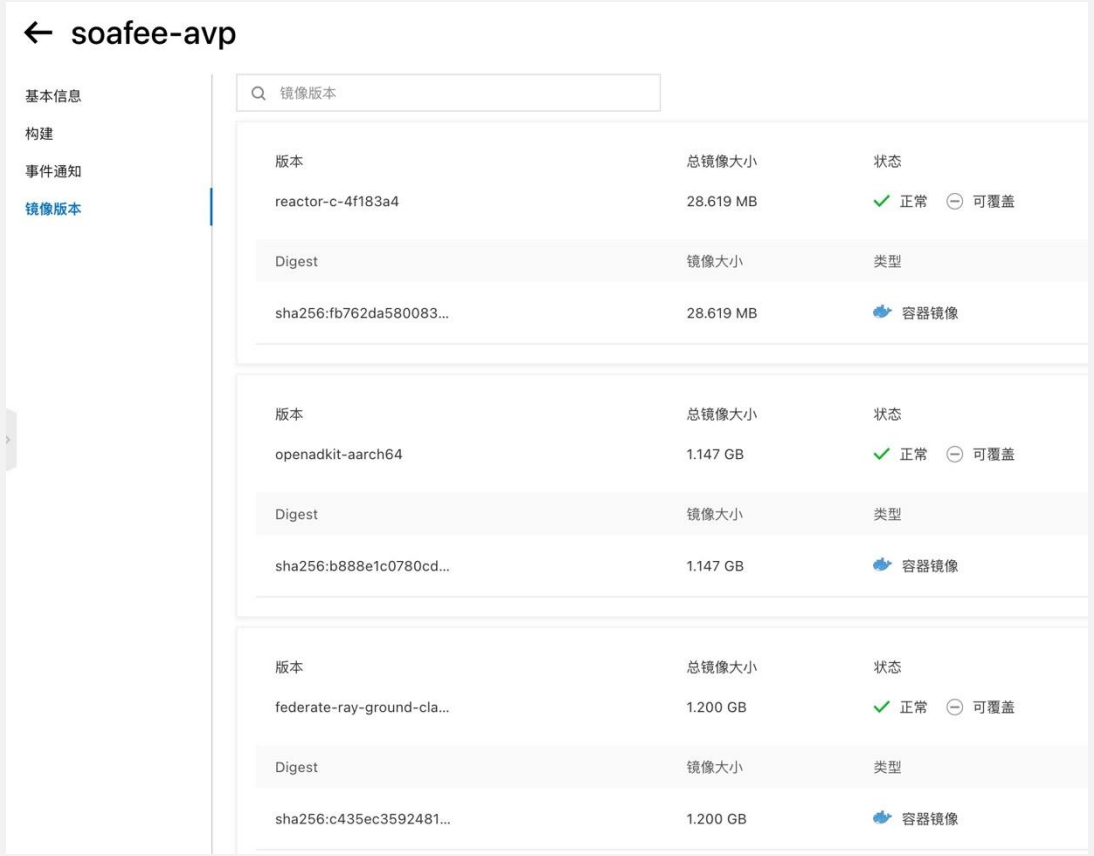
Category	Challenge	Solution
Cloud Infrastructure	Insufficient GPU quota	Early quota & capacity audit; pre-request GPU quota (A10/Tesla) before bulk provisioning
Networking	Network & Dependency Access During Configure / Build	Mirroring, proxy, S3 Asset Re-hosting
Docker Registry	Unable to access DockerHub/ECR images or high latency	Mirrored required images to local container registry



Demo: AVP on China Cloud

Provision, Configure and Build

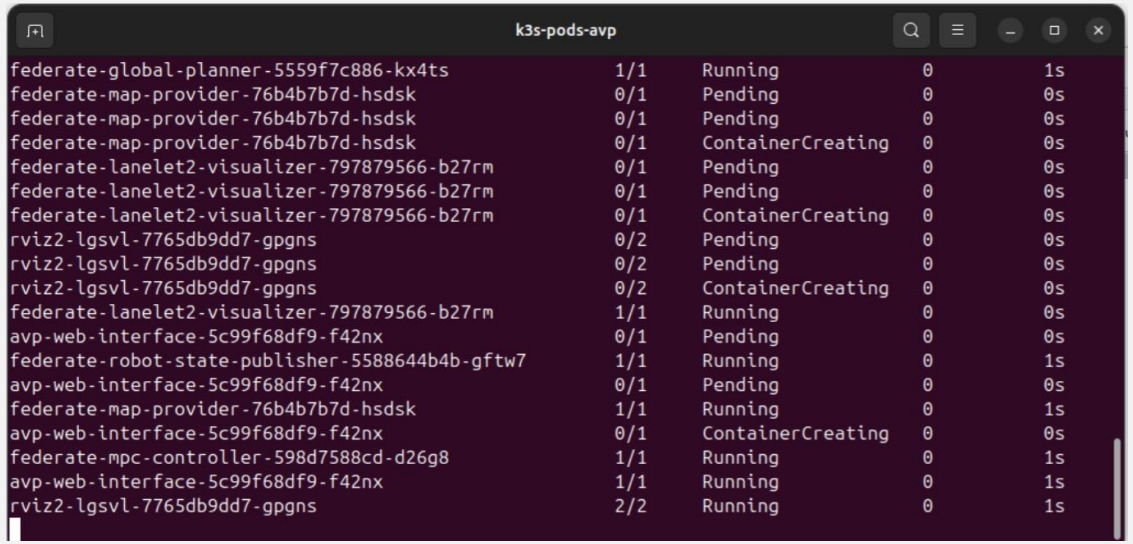


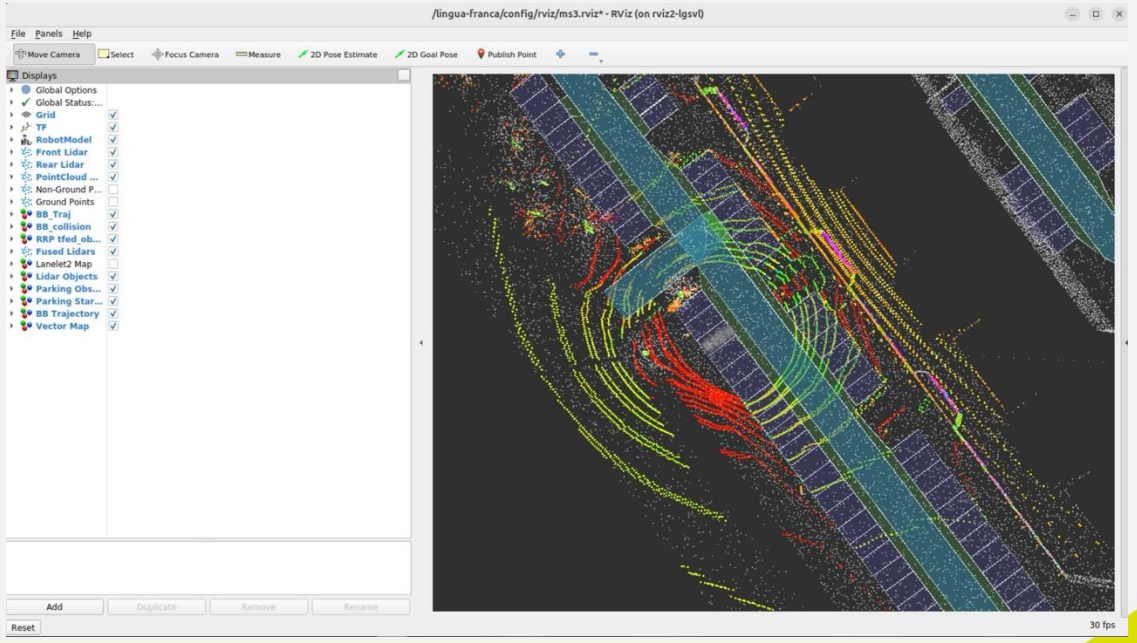


Demo: AVP on China Cloud

Start SOAFEE Blueprint on China Cloud

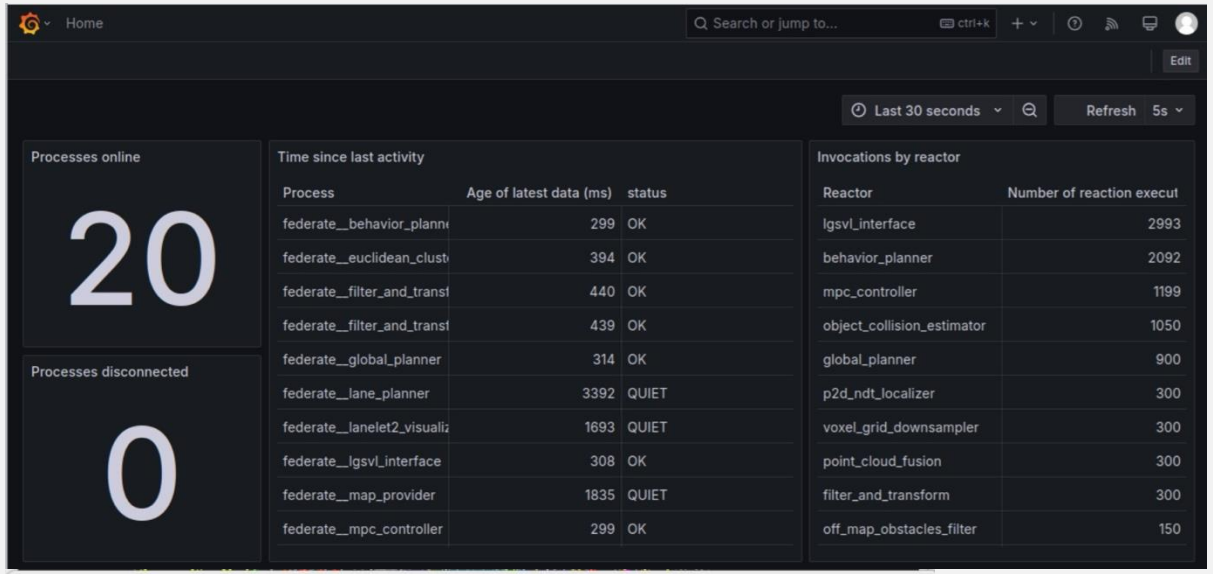
```
weic@ais-wip-ds:~/soafee-avp-blueprint$ ./blueprint start

SOAFEE Blueprint - Autonomous Valet Parking Demo
    by DENSO International America, Inc.

active configuration:
-----------
command:        start
command args:
deployment:     soafee
active user:    avp
active keyfile: soafee-avp.pem
ssh agent:
volumes:        --volume xronos-ansible-cache-roles:/usr/share/ansible/roles
                --volume xronos-ansible-cache-collections:/usr/share/ansible/collections
                --volume xronos-soafee-instances:/home/ubuntu/instances:ro
ansible args:   TASK [start : k3s wait for pods] ***************************************************
                ok: [soafee-avp-render] => (item=rti)
                ok: [soafee-avp-render] => (item=federate-filter-and-transform-0)
                ok: [soafee-avp-render] => (item=federate-filter-and-transform-1)
                ok: [soafee-avp-render] => (item=federate-point-cloud-fusion)
                ok: [soafee-avp-render] => (item=federate-voxel-grid-downsampler)
===> running AVP ok: [soafee-avp-render] => (item=federate-ray-ground-classifier)
                ok: [soafee-avp-render] => (item=federate-euclidean-cluster-detector)
                ok: [soafee-avp-render] => (item=federate-ndt-map-publisher)
                ok: [soafee-avp-render] => (item=federate-p2d-ndt-localizer)
                ok: [soafee-avp-render] => (item=federate-off-map-obstacles-filter)
                ok: [soafee-avp-render] => (item=federate-behavior-planner)
                ok: [soafee-avp-render] => (item=federate-parking-planner)
                ok: [soafee-avp-render] => (item=federate-lane-planner)
                ok: [soafee-avp-render] => (item=federate-global-planner)
                ok: [soafee-avp-render] => (item=federate-object-collision-estimator)
                ok: [soafee-avp-render]
                ok: [soafee-avp-render]
                ok: [soafee-avp-render]
                ok: [soafee-avp-render]
                ok: [soafee-avp-render]
                ok: [soafee-avp-render]
                ok: [soafee-avp-render]

TASK [start : Set websocket bind address 10.43.67.68 in avp-web-interface] *************
changed: [soafee-avp-render]

TASK [start : Open SORA-SVL web interface in Firefox] **********************************
changed: [soafee-avp-render]

TASK [start : Open SORA-SVL docs in Firefox] ***************************************
changed: [soafee-avp-render]

TASK [start : Open avp-web-interface in Firefox] ***********************************
changed: [soafee-avp-render]

TASK [start : Open Grafana in Firefox] *********************************************
changed: [soafee-avp-render]

PLAY RECAP *************************************************************************
soafee-avp-render          : ok=32   changed=12   unreachable=0    failed=0    skipped=3

===> AVP demo command 'start' complete
```

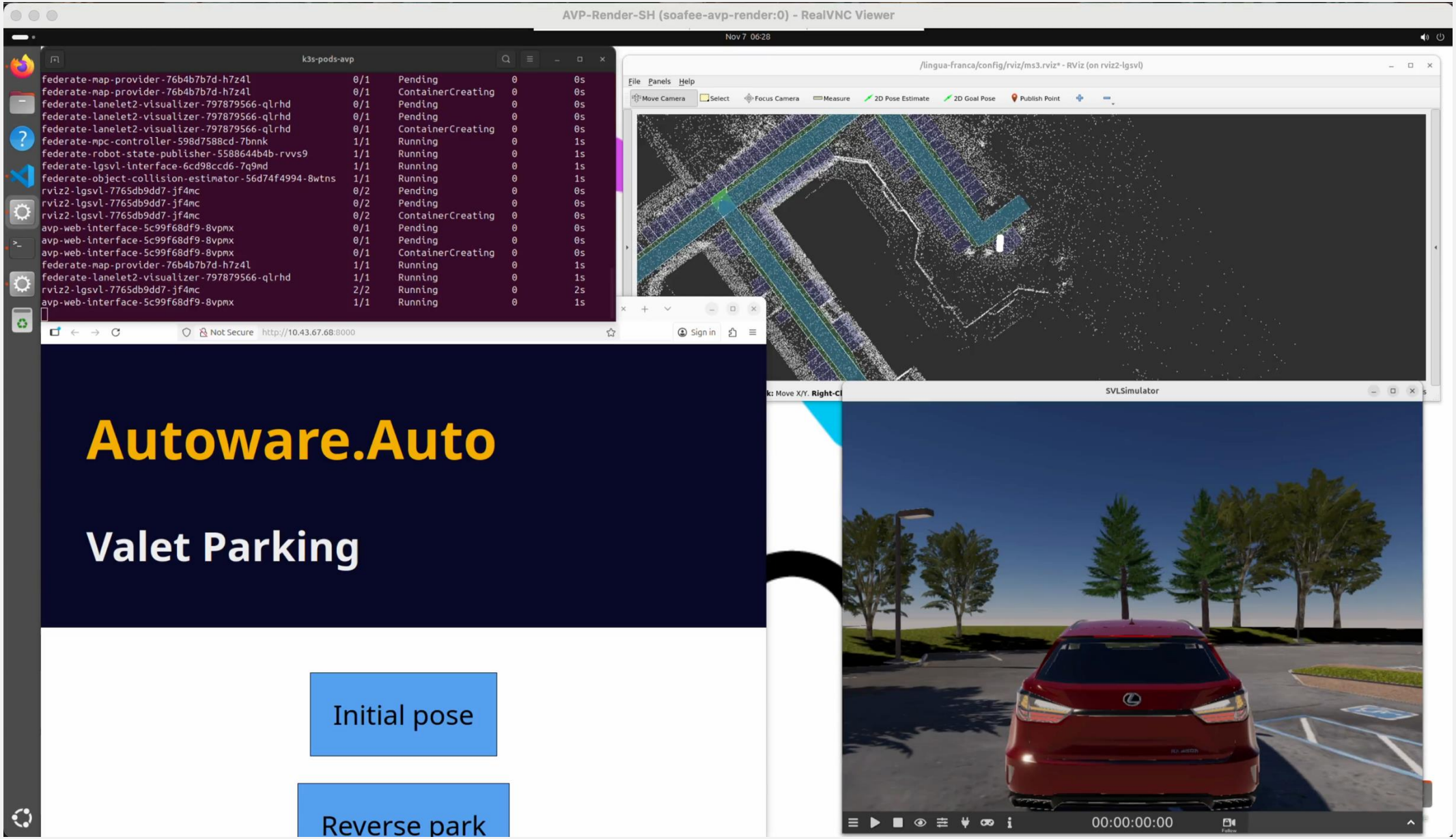




Demo: AVP on China Cloud

Start SOAFEE Blueprint on China Cloud




AVP-Render-SH (soafee-avp-render:0) - RealVNC Viewer
Nov 7 06:28
k3s-pods-avp
federate-map-provider-76b4b7b7d-h7z4l 0/1 Pending 0 0s
federate-map-provider-76b4b7b7d-h7z4l 0/1 ContainerCreating 0 0s
federate-lanelet2-visualizer-797879566-qlrhd 0/1 Pending 0 0s
federate-lanelet2-visualizer-797879566-qlrhd 0/1 Pending 0 0s
federate-lanelet2-visualizer-797879566-qlrhd 0/1 ContainerCreating 0 0s
federate-mpc-controller-598d7588cd-7bnnk 1/1 Running 0 1s
federate-robot-state-publisher-5588644b4b-rvvs9 1/1 Running 0 1s
federate-lgsvl-interface-6cd98ccd6-7q9md 1/1 Running 0 1s
federate-object-collision-estimator-56d74f4994-8wtns 1/1 Running 0 1s
rviz2-lgsvl-7765db9dd7-jf4mc 0/2 Pending 0 0s
rviz2-lgsvl-7765db9dd7-jf4mc 0/2 Pending 0 0s
rviz2-lgsvl-7765db9dd7-jf4mc 0/2 ContainerCreating 0 0s
avp-web-interface-5c99f68df9-8vpmx 0/1 Pending 0 0s
avp-web-interface-5c99f68df9-8vpmx 0/1 Pending 0 0s
avp-web-interface-5c99f68df9-8vpmx 0/1 ContainerCreating 0 0s
federate-map-provider-76b4b7b7d-h7z4l 1/1 Running 0 1s
federate-lanelet2-visualizer-797879566-qlrhd 1/1 Running 0 1s
rviz2-lgsvl-7765db9dd7-jf4mc 2/2 Running 0 2s
avp-web-interface-5c99f68df9-8vpmx 1/1 Running 0 1s
/lingua-franca/config/rviz/ms3.rviz* - RViz (on rviz2-lgsvl)
File Panels Help
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Goal Pose
Publish Point
Not Secure http://10.43.67.68:8000
Sign in
Autoware.Auto
Valet Parking
Initial pose
Reverse park
SVLSimulator
00:00:00:00

Outcomes & Reusable Assets

- Validated Cross-Cloud SOAFEE Abstraction
  - Container orchestration and workload deployment migrated with zero application logic intrusion.
- Repeatable, Compliance-Ready Blueprint
  - A template that accelerates future regional rollouts and shortens localization lead time.
- Localized Image & Automation Supply Chain
  - Enterprise registry mirroring + automated build/sign pipeline improves reliability and reduces cross-border risk.
- Foundation for Multi-Cloud / Edge Collaboration
  - Unified IaC + automation patterns enable future hybrid scenarios and edge node extension.

SOAFEE

Thank You