Anka - macOS CI Infrastructure for iOS/macOS CI

About Veertu

Founded in 2016

HQ in Palo Alto, CA & R&D in Israel

Products – Anka Build, Anka Flow and Anka Scan

Team Background

- Strong background in virtualization, cloud and containers
- Were part of the engineering team that built KVM hypervisor
- Built nested virtualization technology on top of public clouds

Current state of iOS CI Infrastructure setup on physical hardware

Setup	Challenges
Physical mac machines are configured as CI slaves/agents	Difficult to scale – Can’t add slaves/agents on-demand Difficult to keep base macOS system consistent for all builds/test runs to avoid false alerts Difficult to manage testing and rollout of new Xcode release Difficult to manage testing and rollout of new macOS release Parallel job executions causes resource starvation and deadlocks
Scripts an/or imaging is used to baseline the dependencies in the slave physical machine	Introduces possibility of leaving the machine in polluted state with leakage between different job runs Impossible to setup completely isolated instances for each build/test job run Difficult to rollback build/test environment changes

Anka Virtualization to run macOS Vms on macOS (Intel)

New Anka virtualization technology for macOS which uses macOS native hypervisor.framework

- Virtualization application installs on top of macOS and runs as a mac application
- Paravirtual network and block drivers for extreme performance in the guest VMs
- Can create macOS VMs and run (one or multiple) them in a very resource – efficient manner
- Uses macOS native resource scheduling for the VMs
- Instant boots dynamically provisioned macOS VMs
- Virtualizes and runs macOS VMs on all mac hardware
- VM disk space optimization built inside Anka VM
- Command line interface to automate VM creation and bootstrapping process
- Support for Metal (GPU) inside Anka Vm

Anka Virtualization to run macOS Vms on macOS (ARM)

Anka Version 3.0.0 supports creating and running macOS Vms on ARM Mac machines

- Available from Monterey onwards
- Uses Apple's macOS new virtualization.framework
- Supports the existing Anka(Intel) exhaustive command line
- Supports creating a private macOS cloud using Anka Build Cloud components – Controller and registry
- Supports cloning of VM images
- Supports advanced networking features for the VM like port forwarding, bridge etc

Anka Products

Anka Build

To configure a macOS cloud on a cluster of mac hardware running Sierra or higher

Anka Flow

To enable developers to build and test in consistent iOS dev environments locally on their development machines

Anka Scan

To execute vulnerability scanning on macOS Anka images used for iOS CI jobs

Supported on mac Hosts in Data centers, hosting providers and AWS EC2 Mac instances

macOS Anka Build cloud components

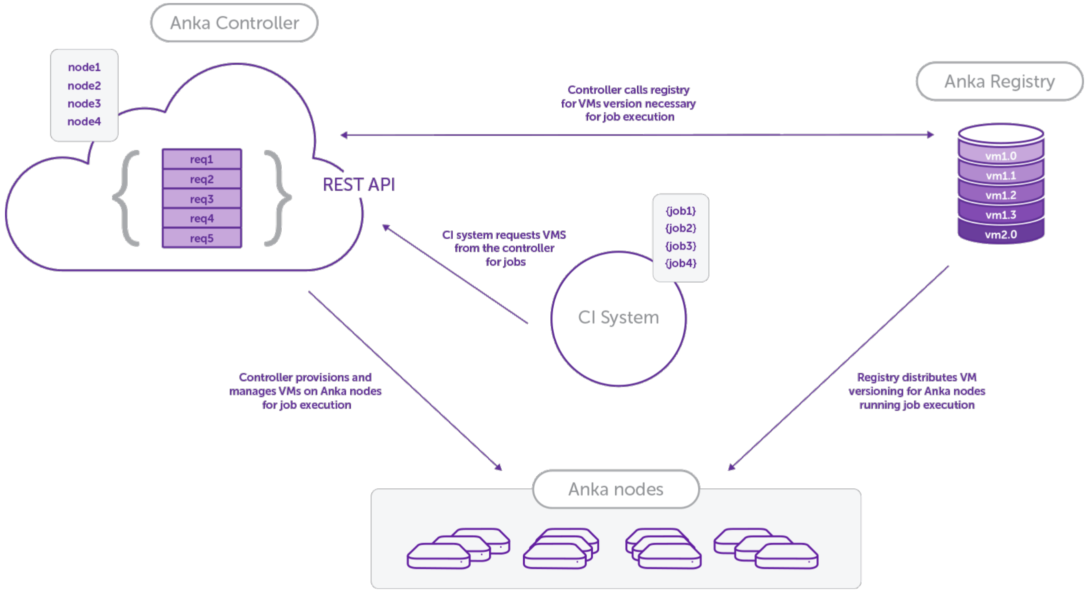
- Controller - Controller is the cloud management software module, which exposes provisioning REST APIs and manages scheduling of macOS VMs on the mac cluster. Available as a docker Linux container.

- Registry – Registry is the central repository to store, version manage and distribute macOS VM templates across the mac cluster. Available as a docker Linux container.

- Anka Nodes – Mac machines which are running the Anka virtualization software `Anka Build.pkg`. These nodes run the Anka macOS on-demand VMs.

- Jenkins Plugin – Plugin available through Jenkins plugin center to configure Anka cloud inside Jenkins.

- TeamCity Plugin – Plugin available through Teamcity Plugin repository.


Anka Controller
node1
node2
node3
node4
req1
req2
req3
req4
req5
REST API
Controller calls registry for VMs version necessary for job execution
Anka Registry
vm1.0
vm1.1
vm1.2
vm1.3
vm2.0
CI system requests VMS from the controller for jobs
{job1}
{job2}
{job3}
{job4}
CI System
Controller provisions and manages VMs on Anka nodes for job execution
Registry distributes VM versioning for Anka nodes running job execution
Anka nodes

Creating macOS VMs in Anka

Supports creation from .app installer, ISO and AutoDMG

- Can fully automate VM creation using `anka create` command and .app installer.
- Then, use `anka run` similar to docker run to bootstrap VM with all dependencies/packages.

Managing VM templates with Anka Registry

Similar to Docker Registry – Store, version and distribute VM templates for dynamic provisioning on Anka Build cloud or local development with Anka Flow

-> Incremental pull and push

anka registry push –d "first version" –t v1 Hisierra
anka registry push –d "second version" –t v2 Hisierra

anka registry pull Hisierra (pulls latest tag)
anka registry pull – v1 Hisierra (pulls tag V1)

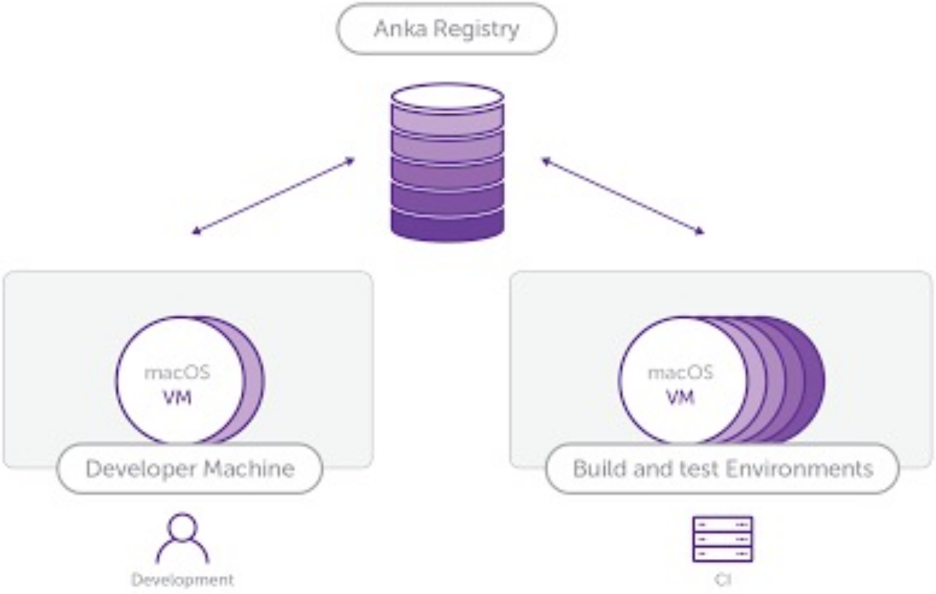
Solutions and Use Cases

Anka Flow – Consistent environments for developer machines
Modules included – Anka Flow.pkg

Developers pull build/test VMs from central CI Anka registry on their machines

Use anka run(docker run) interface and shared folder to build and test inside the Anka VMs from their main dev machine.

Eliminates *"But, it runs on my machine"* issue and reduces failures upstream in the CI workflow



Solutions and Use Cases

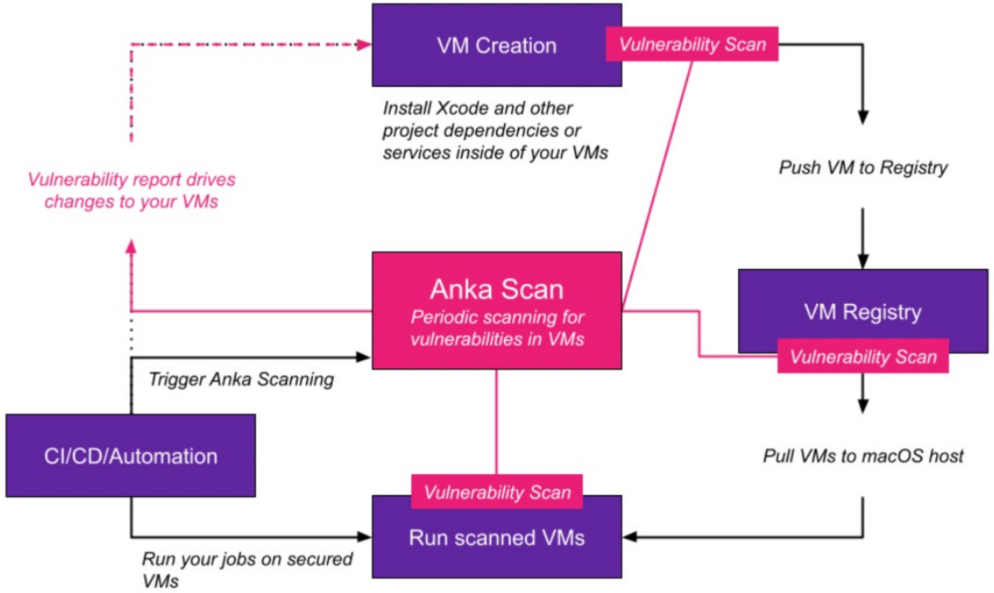
Anka Scan – Execute continuous vulnerability scanning on iOS CI build and test macOS VM Images

Identify known vulnerabilities associated with all packages, libraries, agents on macOS filesystem against an NVD database.

At-rest macOS VM scanning to minimize complexity and effort in scanning for vulnerabilities.

Vulnerability scanning support for Intel or ARM macOS VMs.

Continuously scan VMs in your registry for newly discovered vulnerabilities.



VM Creation
Vulnerability Scan
Install Xcode and other project dependencies or services inside of your VMs
Push VM to Registry
Vulnerability report drives changes to your VMs
Anka Scan
Periodic scanning for vulnerabilities in VMs
VM Registry
Vulnerability Scan
Trigger Anka Scanning
CI/CD/Automation
Pull VMs to macOS host
Vulnerability Scan
Run scanned VMs
Run your jobs on secured VMs