CVB Installation Doc for R2

Installation Architecture

Due to the hardware source limitation, Connected Vehicle Blueprint is deployed in three Virtual Machines in Amazon Web Service.

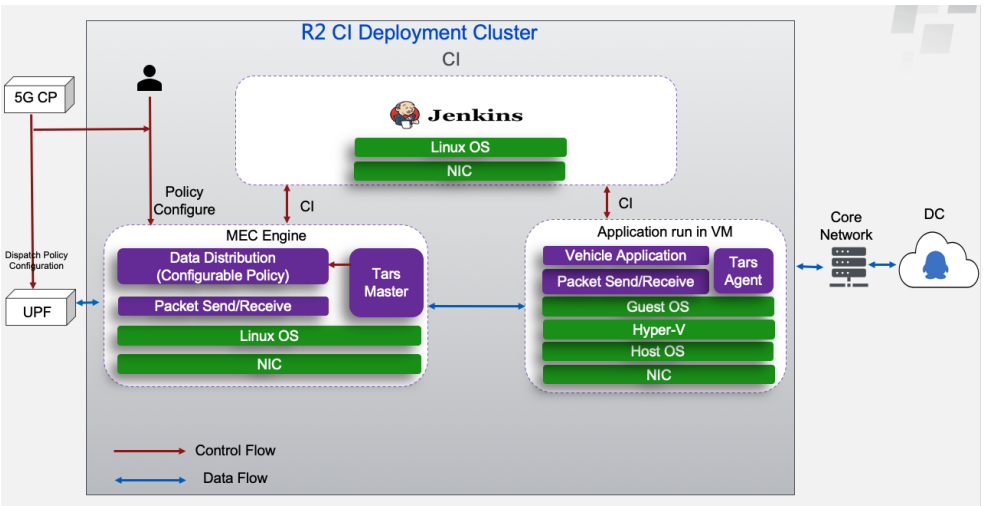
(Visit CI Lab Environment Setup if you want to set up connected vehicle blueprint in CI Lab.)

The following picture depicts the deployment architecture.

Server A ：Deploy Jenkins.

Server B ：Deploy Tars Master Node.

Server C Deploy Tars Slave(Node) and the connected vehicle applications.



Installation Hardware

Server Name	CPU+Memory	Drive	Deployment
Jenkins	A1 8Core * 16G	15G	Jenkins Master
TarsFramework	A1 8Core * 16G	10+50G	TarsFramework
TarsNode	A1 8Core * 16G	10G +20G	TarsNode + Application

Installation Software

- CentOS 7 centos-7_aarch64 - ami-012355fc520b79a12
- mysql Ver 14.14 Distrib 5.6.26, for Linux (aarch64) using EditLine wrapper
- OpenStack: Rocky
- k8s:1.15.0

Installation Step by Step

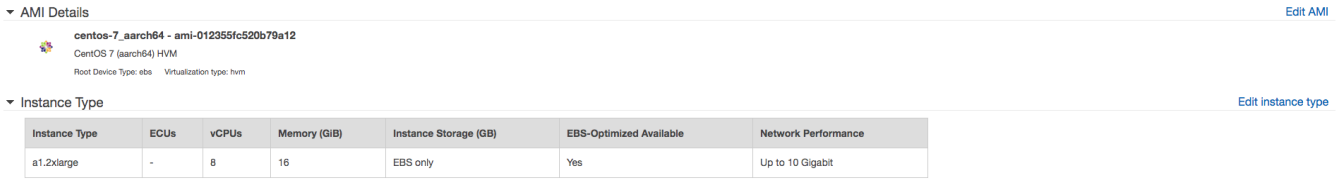
Step1: Launch Instance in the Cloud

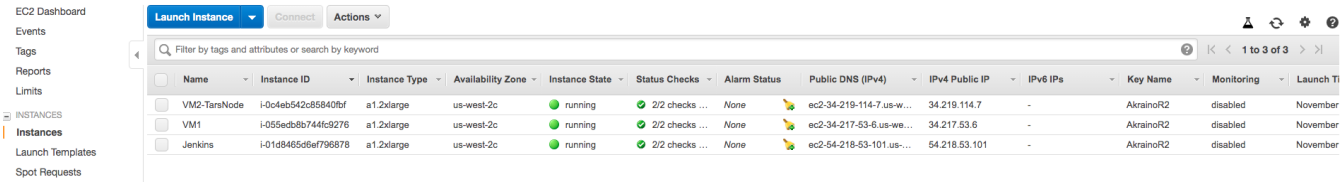
For Tencent Cloud, refer to the following link to apply new instance:

https://intl.cloud.tencent.com/document/product/213/9384?lang=en

For AWS A1, apply new instance, refer to the following link to apply new instance:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html

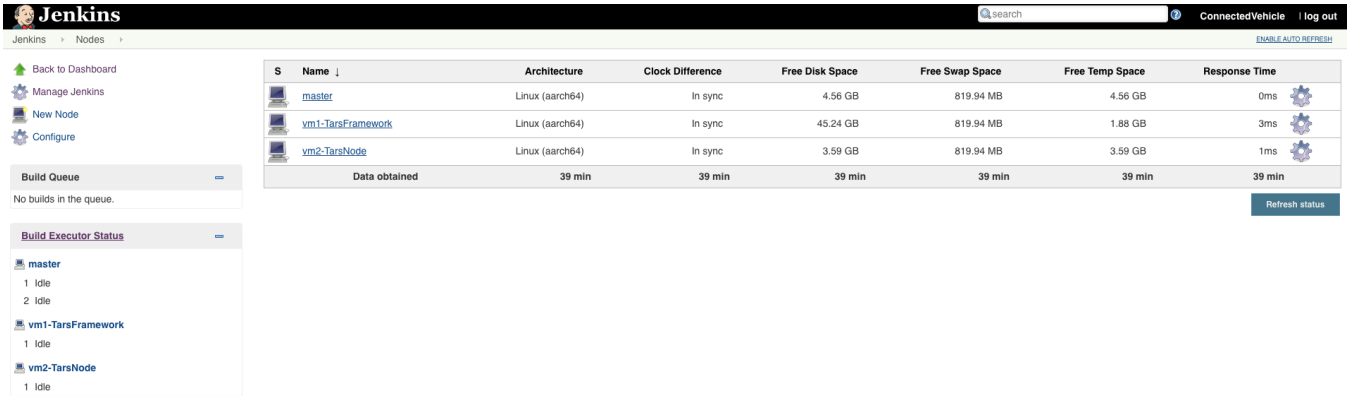


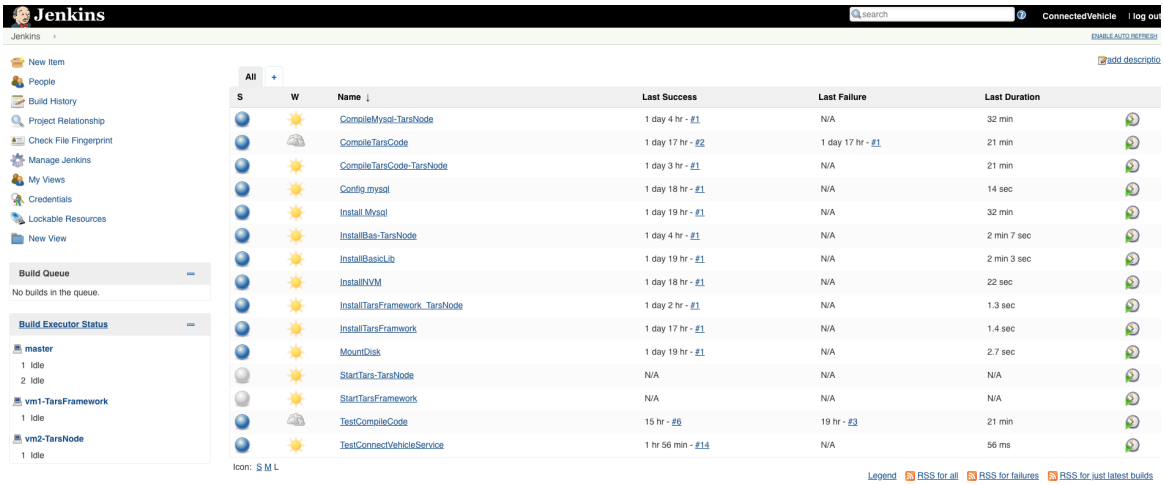


Step2: Install Jenkins

Refer to the following link for installing Jenkins Mater and connect to Jenkins Slave.

https://github.com/qiuxin/Connected-Vechicle



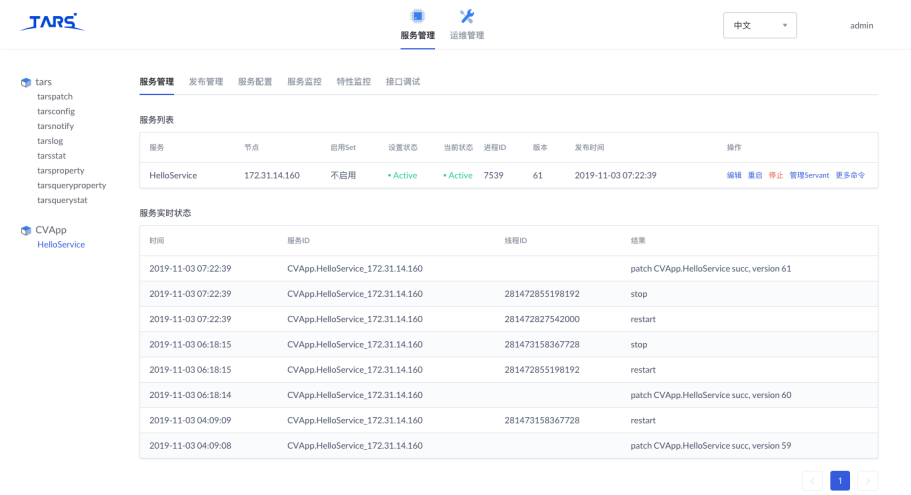


Step3: Install Tarsframework

Refer to the following link for installing Tarsframework.

https://github.com/qiuxin/Tars/blob/arm/Install.md

The following is the picture for Tars Framework website.



Step4: Tars Node

Refer to the following link for installing Tarsnode and connect Tarsnode to Tarsframework.

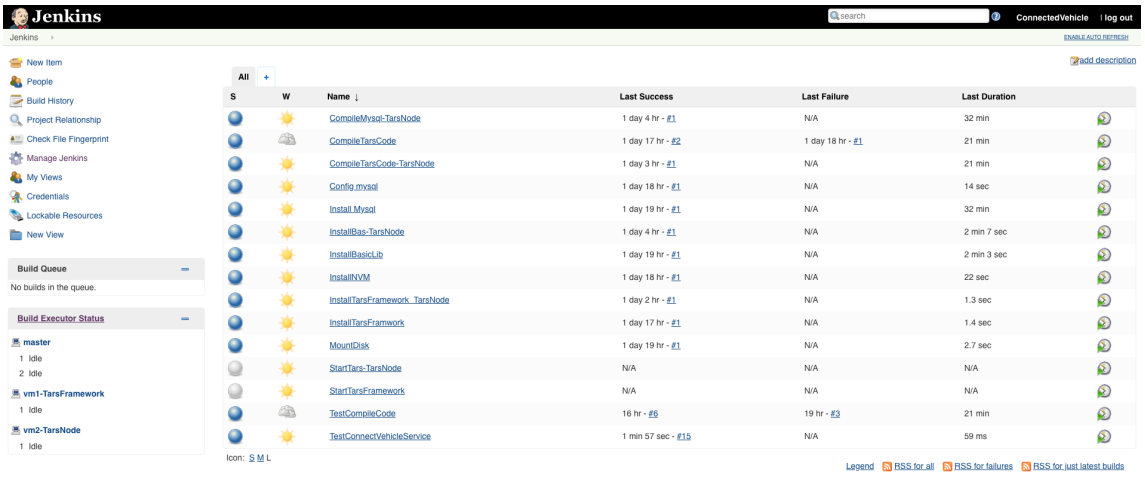
https://github.com/qiuxin/Tars/blob/arm/Install.md

Step5: Launch CI jobs

Create CI jobs and launch CI jobs in the following way.

The detail of the Jenkins script is depicted in the Test Document.

CVB Test Doc for R2



Jenkins › TestCompileCode ›

General | Source Code Management | **Build Triggers** | Build Environment | Build | Post-build Actions

- [ ] Build after other projects are built
- [x] Build periodically

Schedule

```
0 12 * * *
```

⚠ Spread load evenly by using 'H 12 * * *' rather than '0 12 * * *'
Would last have run at Sunday, November 3, 2019 12:00:08 PM UTC; would next run at Monday, November 4, 2019 12:00:08 PM UTC.

- [ ] GitHub hook trigger for GITScm polling
- [ ] Poll SCM

Build Environment

- [ ] Delete workspace before build starts
- [ ] Use secret text(s) or file(s)
- [ ] Abort the build if it's stuck
- [ ] Add timestamps to the Console Output
- [ ] Inspect build log for published Gradle build scans
- [ ] With Ant

Build

Execute shell

Command

```
cd /usr/local/robert/testCompileCode
git clone -b arm https://github.com/qiuxin/Tars.git
cd /usr/local/robert/testCompileCode/Tars
git submodule update --init --recursive
cd /usr/local/robert/testCompileCode/Tars/framework/build
chmod u+x build.sh
./build.sh all
rm -rf /usr/local/robert/testCompileCode/Tars
```

See the list of available environment variables

Save | Apply | Advanced...