* Local.conf file is basically used to customize the image
* We can add packages that we need in the image also here
* So basically yocto build linux distributions for various embedded boards like beagle bone, raspberry pi, s32g2 board from nxp etc
* So in the local.conf file we can set the machine to which board we use
* For boards like nxp or Texas instruments, they provide a meta-layer or a layer in which the machines will be available
* When you set a certain machine in local.conf it means that u this board has certain configurations that yocto will set for you
* We have a SOURCES variable in local.conf file, this folder will have 3 folders inside it
  + DL\_DIR
  + SSTATE\_DIR
  + TMPDIR
* DL\_DIR is the download folder where all the souce code will be downloaded so that next time we wouldn’t have to download it again
* SSTATE\_DIR specifies where shared state cache files are stored, means this folder contains the pre-built binary packages for sw components, say a pkg is built then its binaries will be saved in SSTATE CACHE. The benefit is if we didn’t make any changes in our source files, and if we build the image again, then it wont compile again instead it will take the binary from sstate\_cache
* TMP\_DIR contains the temporary files used by various programs and scripts (This is where we find our binaries, in tmp/deploy/image\_name
* DISTRO – This specifies the name of the distribution that is built, a distribution is a collection of software components and configuration files that work together to complete an OS, in the file we have set it to poky. So poky is also a distro similar to ubuntu or fedora, but when we say distro it means it is a full fledged os with pkg managers, user interfaces.
* Poky is not like that, we should call it a **BUILD SYSTEM**  that helps create a custom linux distro
* PACKAGE\_CLASSES variable is used to used to set the type of package we want to create for the target system, it can be deb or rpm
* In vpc-p also (Have some shame bitch) But I need to analyze how this rpm package is being used(check in build\_vpc-p\_(image)/conf/local.conf
* PACKAGE\_CLASSES ?= "package\_rpm"
* EXTRA\_IMAGE\_FEATURES is used to set additional features to be included in the target image, in the example it is set to debug-tweaks, this is used for debugging

EXTRA\_IMAGE\_FEATURES ?= "debug-tweaks"

* USER\_CLASSES variable is used to apply additional processing to your image, think of it as optional plugins that can be enabled during build. In the example buildstats is used to record the performance statistics of each task executed and it starts executing during the build process
* IN a build process the first task is to fetch, upacks, then configure and then patch and install and compile, in bb file like we write do\_compile, do\_fetch, do\_install, do\_configure, do\_patch, do\_remove etc
* USER\_CLASSES ?= "buildstats"
* PATCH\_RESOLVE can take 2 values noop or user. Say I apply a patch to a file and then there is an error occurred during compiling, then when I set it to noop then the build will fail because the patch was not applied properly, if I set PATCH\_RESOLVE to user then if patch fails, it will open the terminal and tell us to resolve this issue manually
* PATCHRESOLVE = "noop"
* BB\_DISKMON\_DIRS – This variable is very important and it should always be present in the local.conf file . This is a part of the **BitBake disk monitoring system**
* STOPTASKS,${TMPDIR},1G,100K \
* STOPTASKS,${DL\_DIR},1G,100K \
* STOPTASKS,${SSTATE\_DIR},1G,100K \
* Here the STOPTASKS specifies If the disk space used by ${TMPDIR} or {DL\_DIR} or {SSTATE\_DIR} exceeds **1 GB**, stop tasks.
* If the available disk space drops below **100 KB**, stop tasks immediately.
* HALT,${TMPDIR},100M,1K \
* HALT,${DL\_DIR},100M,1K \
* HALT,${SSTATE\_DIR},100M,1K \
* HALT,/tmp,10M,1K"
* Halt command will stop the build completely if either disk space exceeds 100M or available disk space drops below 1KB
* Unlike STOPTASKS, which only stops specific tasks, HALT stops bitbake entirely
* CONF\_VERSION is used to specify the version of the configuration used