# Git basics

## මොනවද මේ version controlling කියන්නේ?

අපි ගාව මොකක්ම හෝ වර්ගයේ වැදගත් file එකක් හෝ කීපයක් තියෙනවා කියල හිතමු (text file එකක්, codes තියෙන file එකක්, image edit කරන file එකක් වගේ ඕනෑම file එකක් හෝ කීපයක්).නමුත් මේවා අපි edit කරනවා නම් සමහර විට යම් යම් වැරදි අපි අතින් සිද්ද වෙන්න පුළුවන්.එහෙම වැරද්දක් වෙලා තියෙන විට අපි ඒක save කරොත් එහෙම යලිත් පළමු තත්වට ගන්නට සෑහෙන වෙහෙසක් ගන්න වෙනවා.

මෙන්න මේ වෙලාවට තමා **version controlling systems** එහෙම නැත්තම් **version controlling පද්දති** වැදගත් වෙන්නේ.

මේ පද්දතියක් විසින් කරන්නේ අපි යම් file එකක් හෝ කීපයක් මත සිදු කරන වෙනස් කම් මොනවද කියල track කරන් (සටහන් කරගෙන) ඉන්නවා.එතකොට අපිට ගොඩක් ඉස්සර දවසක, file එක තිබ්බ තත්වයකට, ගන්න ඉතාමත් ලේසියෙන් පුළුවන් වෙනවා.

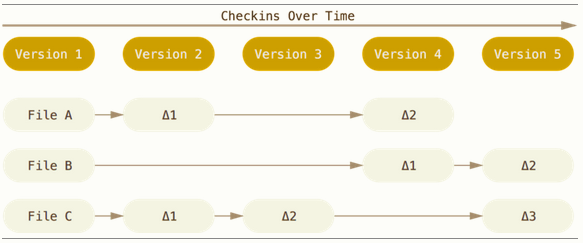
file එකකට යම් වෙනසක් කල විට, එම වෙනසක් සිදු වූ file එකට කියන්නේ, පරණ file එකේ අලුත් **version** එකක් කියලයි.අන්න එහෙමයි ඔය version controlling කියන නම ආවේ.

## හරි, එතකොට මොකක්ද git කියන්නේ?

git කියන්නේ ඉහතින් කිව්ව වගේ version controlling කරන පද්දතියක්.එනම් files වල changes track කරන් ඉන්න පුළුවන් පද්දතියක්.

ඔව්,තව මේ වගේ version controlling systems තියෙනවා.නමුත් ගොඩක් ජනප්‍රිය වුනේ මේ git කියන පද්දතිය.එකට හේතු කීපයක් තියෙනවා.එකක් තමා ඒක complete system එකක්.ගොඩක් වැඩ කෑලි තියෙනවා කරන්ඩ පුළුවන්.අනෙක් එක තමා git පද්දතිය version controlling කරන්ඩ භාවිතා කරන ක්‍රමය අනෙක් version controlling systems වලට වඩා වෙනස්.මේ ක්‍රමය අපි හොදින් තේරුම් ගන්න ඕන.ඒක පහලින් විස්තර කරලා තියෙනවා.

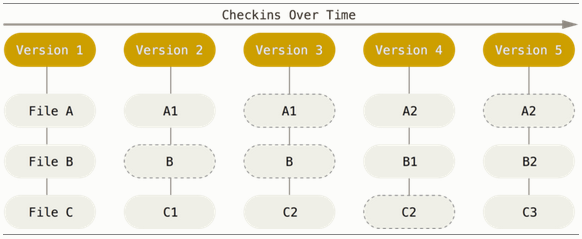
## වෙනස්කම් වෙනුවට photos

git සහ අනික් VCS අතර තියෙන ප්‍රදානම වෙනස තමයි, git එයට අදාළ data (track කරන් ඉන්න කියල භාර දීලා තියෙන files) ගැන හිතන විදිය. ගොඩක් අනික් පද්දති store කරගන්න තොරතුර වන්නේ files වල සිදු වු වෙනස කුමක්ද කියන එකයි.එනම් ඒ ඒ file එකේ සිදු වු වෙනස්කම් මොනවද කියල එම වෙනස් කම සිදු වූ වෙලාවත් සමග list එකක වගේ store කරන ඉන්නවා එම VCS එකේ database එක තුල.පහතින් දැක්වෙන්නේ ඒක illustrate කරන ෆොටෝ එකක්.

ඉහත රූපයේ File A, File B, File C කියල files තුනක් තියෙනවා. මේවා version controlling පද්දතියට හදුන්වා දුන්නහම මෙන්න මේවා ගැන පොඩ්ඩක් ඇහැ ගහගෙන ඉන්න කියල, එවිට පද්දතිය මෙම ආරම්බක files ටික සලකන්නේ මේවා තමා version 1 files ටික කියල.

ඉන් පසුව අපි files වලට යම් වෙනසක් කළහොත්, git හැර අනෙක් බොහෝ පද්දති මෙන්න මේ වෙනස පමණක් store කරගන්නවා එය සිදු වූ වෙලාවත් සමග එහි database එකේ.මෙන්න මේ **වෙනසයි** එම පද්දති දෙවන, තුන්වන versions ලෙස සලකන්නේ (ඒකයි ඉහත ෆොටෝ එකේ delta ලකුණ ඉදිරිපත් කරලා තියෙන්නේ වෙනස්කම් ඉදිරිපත් කරන්න) .so මෙම වෙනස් කම් මොනවද කියල දන්නා නිසා, ඒවා undo කර තමන් කැමති version එකක් දක්වා file එක නැවත පත් කරන්න පුළුවන්.

Git එයට අදාළ files (data) ගැන හිතන්නෙත් , database එකේ store කරගන්නෙත් ඉහත ක්‍රමයට නෙමේ. Git එයට අදාළ data ගැන හිතන්නේ photos ටිකක් ලෙසයි.අපි commit එකක් කල විට (වෙනසක් සිදු කර, ඒ බව ස්ථිර කල විට), git පද්දතිය කරන්නේ එම files ටිකේ වර්තමාන තත්වයේ photos ටිකක් ගෙන එම photos ටිකට අදාළ reference එකක් එහි database එකේ save කරගැනීමයි.වඩාත් කාර්යක්ෂම වීම සදහා, යම් files කීපයක් වෙනස් වී නොමැති නම්, එවිට පරණ එම වෙනස් නොවූ files ටිකට link වන links ටික පමණක් database එකේ store කරගනියි එම version එකේ එම files සදහා. එම නිසා git එයට අදාළ data ගැන හිතන්නේ ෆොටෝ වලින් සැදුනු ෆොටෝ වැලක් ලෙසයි.එක ෆොටෝ එකක් යනු එක version එකක් වේ.පහත රුපයෙන් එය පැහැදිලි කරයි.මෙහි කඩ ඉරි යනු modify නොකෙරු files වේ.



## හැම operation එකක්ම ගොඩක් වෙලාවට local

git වල තියෙන ගොඩක් operations වලට (වැඩ කෑලි වලට) අවශ්‍ය වන්නේ අපේ computer එකේ තියෙන files සහ උපාංග විතරයි.CVCS ගණයට අයත් වන version controlling පද්දති වලට එහෙම බෑ.ඒවාට internet එහෙම නැත්තම් network පහසුකම් අවශ්‍ය වෙනවා.CVCS කියන්නේ Centrallized version Controlling systems කියන එකටයි. Git අයිති වන්නේ DVCS කියන ගණයට.එනම් Decentrallized version controlling system ගණයට. CVCS සහ DVCS කියන්නේ මොනවද කියල පස්සේ කතා කරමු.

මෙනිසා git ගොඩක් speed.

## Git ආරක්ෂිතයි

git වලදී store කරන හැම දෙයකම check-sum අගය ගනයන කර ඉන් පසුවයි store කරගන්නේ. ඉන් පසු එම store කරගත් දේවල් access කරන්න භාවිතා කරන්නේ එම check-sum අගයයි.එම නිසා git වලට කියන්නේ නැතුව කිසිවක් change කරන්ඩ කවදාවත් බෑ. check-sum එක ගණනය කරන්ඩ git භාවිතා කරන ක්‍රමය තමා SHA-1 කියන hash function එක.මේක characters 40ක් දිග string එකක්.git data store කරගනිද්දී එහි database එකේ, store කරගන්නේ එම version එකට අදාළ files වල check-sum එක මෙම hash function එකෙන් ගණනය කර එන, ඉහත කියන ලද characters 40 ක් දිග string එකේ නමිනුයි.

ඉහත කියපු ටික තේරෙන් නැති වුනාට අවුලක් ගන්ඩ ඕන නෑ.අවශ්‍යනම් වැඩි දුර විස්තර හොයන කෙනෙකුට ලෙසි වෙන්ඩ ඕන නිසයි දැම්මේ.SHA-1 hash function එක ගැන ටිකක් හොයල බලන්න අවශ්‍ය නම්. නමුත් git වලට එම function එක ගැන දැනුමක් කොහෙත්ම අවශ්‍ය වන්නේ නෑ.

දැනගන්න ඕන git පද්දතිය ගොඩක් secure කියල විතරයි.

## **The Three States (අවස්ථා තුනයි)**

මේ තමා git ගැන මතක තියාගන්න ඕන සහ තේරුම් ගත යුතු වැදගත්ම දේවල් වලින් එකක්.So, pay attention. Git පද්දතිය සතුව අවස්ථා තුනක් තියෙනවා අපේ files ටිකට තියෙන්න පුළුවන්.එම අවස්ථා තුන තමා

1.commited අවස්ථාව,

2.modified අවස්ථාව,

3.staged අවස්ථව

commited කියන්නේ,අපි file එකක වෙනස් කම් කරලා,එම වෙනස් කම සහිතව file එකේ snapshot එකක් git database එකේ store කරලා තියෙන අවස්ථාව.

modified කියන්නේ,අපි file එකක් වෙනස් කරලා, නමුත් ඒවා git database එකට commit කරලා නෑ.එතකොට ඒ file එක තියෙන්නේ modified කියන අවස්ථාවේ.

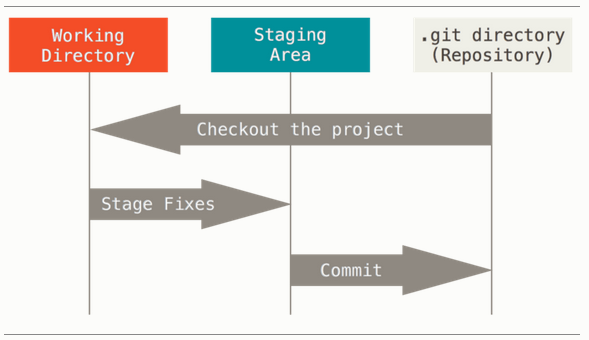
Staged කියන්නේ,අපි වෙනස් කරපු file එකක් commit කරන්ඩ කලින් ඒක තොරගන්ඩ ඕන.එලස තෝරාගත් files ටික විතරයි commit කියන විදානය දුන්නහම database එකේ store වෙන්නේ.එනම් අපේ folder එකක් ඇතුලේ files 10ක් තියෙනවා කියල හිතන්න.අපි මෙහි files 7ක වෙනස් කම් සිදු කරනවා. එවිට එම files 7ම modified අවස්ථාවට පත් වෙනවා.commit කරන්න කලින්, අපි තෝරාගත යුතුයි මෙම files 7 අතුරින් කුමන files ටිකද commit කරන්නේ කියල.එනම් කුමන files ටිකද ෆොටෝ එකට (snapshot) එකට අහුවෙන්න ඕනා කියල.මෙලස යම් file එකක් තෝරාගත්තොත් ඒක staged අවස්ථාවට පත් වෙනවා. (අනික් VCS වල මෙහෙම state එකක් නෑ. git වල මෙහෙම එකක් තියෙන නිසා git ගොඩක් නම්‍යශීලී පද්ද්දතියක් කියල ගොඩක් අය කියනව)

මෙම Staged කියන අවස්ථාව තේරුම් ගන්න සරල උදාහරණයක් තියෙනවා.එනම් මෙම Staged කියන state එක නැව් වරායක් කියල හිතන්න පුළුවන්.අපිට අවශ්‍ය භාණ්ඩ ගොඩබිමේ හදනවා. export කරන්ඩ සුදුසු තත්වයේ හා අපි export කරන්න කැමති ඒවා පමණක් deck (වරායට) ගන්නවා (Staged අවස්ථාවට පත්කරනවා). ඉන් පසු export කරනවා (commit කරනවා).

මේ එක එක state එකක් බවට පත් කිරීම සදහා git භාවිතා කරන command ටිකක් තියෙනවා.ඒවා ගැන පස්සේ බලමු.

ඉහත හදුන්වනු ලැබූ state තුනත් සමග දැන් අපිට කතා කරන්න පුළුවන් git පද්දතියේ ඉහත කියන ලද state තුනට අදාළ වන sections ගැන.states තුනක් තියෙන නිසා ඒවාට අදාළ වන sections එහෙම නැත්තම් places ද තුනක් තියෙනවා.ඒවා තමා

1. Git directory,
2. Working directroy සහ
3. staging area



Git directory (ගොඩක් අය කියන්නේ මේකට git repository කියල) එකේ තමා git පද්දතියට අදාළ database එක හා එයට අදාළ metadata තියෙන්නේ.ඉතින් මේක තමා වැදගත්ම කොටස git වල. අපි වෙන computer එකකින් හෝ server (git hub වගේ) එකකින් repository එකක් එහෙම නැත්තම් project එකක් clone කරහම (අරගත්තහම) අපේ පරිගණකයේ copy වෙන්නේ මෙම කොටසයි.

අපි වැඩ කරන් යන directory එක මෙය වේ.මෙහි තියෙන files තමා අපි edit කරන්නේ.අපි වෙන තැනකින් repository එක import කරගතහොත්, එවිට එන්නේ ඉහත කිව්වා වගේ repository එක සහිත compressed වුන file එකකි.එය extract කල විට අපිට එම project එකට අදාළ working directory එක ගන්න පුළුවන්.

staging area කියන්නේ ඇත්තටම තනි file එකකටයි.. ඒක තියෙන්නේ git repository එකේය. එම file එකේ සදහන් වන්නේ ඊලග commit එකට අදාළ වන දේවල් මොනවද කියලයි.එනම් මේක තමා අපේ උදාහරණයේ විදියට නම් වරාය.මේකට සමහර අය git වල index file එක කියලත් කියනවා.

### git වලදී අපි සාමාන්‍යයෙන් සිදු කරන ක්‍රියා පිළිවෙල පහත පරිදි වේ.

* working directory (folder) එකේ තියෙන files ටිකක් modify කිරම.
* එම files stage එකට ගැනීම.
* commit කිරීම. stage එකේ තියෙන files වල snapshot එකක් අරන් එය git directory එකේ store කිරීම. stage එකේ නොමැති files වෙනුවට එම snapshot එකේ තියෙන්නේ එම files ටිකට reference එකක් වේ.

ඊලග article එකෙදී අපි බලමු කොහමද git install කරගන්නේ කියල හා එහි basics configurations ටික කරන්නේ කොහමද කියල.