Phases	Source	Steps	Owner	Jira	Actions	Comments	
1	Planning	Bitbucket, Existing GitHub Account	Users list with Permissions for the repositories to add users to respective groups	Kraken			Permission Types Read: Recommended for non-code contributors who want to view or discuss your project Triage: Recommended for contributors who need to proactively manage issues, discussions, and pull requests without write access Write: Recommended for contributors who actively push to your project Maintain: Recommended for project managers who need to manage the repository without access to sensitive or destructive actions Admin: Recommended for people who need full access to the project, including sensitive and destructive actions like managing security or deleting a repository
2		Bitbucket	List Down all the repositories which we need to migrate	Kraken		List All Repositories Identify Owner repository to define Custom Property (Currently its Owner) which we will setup to repository during Migration	In this process we have opportunity to archive unused repositories and Branches We can create a priority list which we will migrate first
3		GitHub	All GitHub Settings and configuration	Kraken		TODO in Progress	
4	Initial Setup	Azure AD	Create Centralize User account to Create GitHub Enterprise	IT team			
5		GitHub	Create Enterprise account	Kraken		Create Enterprise Account and connect replace me Organization	
6		GitHub	Define and Set Org level Permissions Repository Rules Commit Control Settings	Kraken			
7		GitHub	Define and set Member privileges	Kraken			
8		Azure AD	Configure Enterprise Organization SSO application	Kraken			
9		GitHub	Configure SSO	Kraken		Enabled Force use of SSO for Organization after Testing is completed	
10		Azure AD	Enable Auto Provisioning for enterprise application	Kraken			Function to add and remove users automatically from GitHub after user is disabled/deleted from azure ad
11		Azure AD	Create Groups and attach it to Enterprise application	IT team			
12		GitHub	Create Team/Groups and connect to azure identity provider groups	Kraken			
13		Azure AD	Create User or attach existing user to group	IT team			Create user with following mandatory fields in azure, Name, FirstName, LastName
14		Migration	Create Centralize User account for Kraken Activities	IT team and Kraken		Add SSH Key (Buildkite agents, Terraform Cloud) Create Taken as per required permissions (TeamCity)	
15		Jira Integration	Integrate Jira			Use Centralize account to connect to Jira with GitHub If Centralize account is not available on Jira create one with permissions	
16		Review Setup	Validate all infosec Requirements	Infosec		Infosec Team will review all the settings Will Confirm if meeting all required conditions Review if any changes suggested	
17	**Dry Testing**						
18	**Pre Migration**	Developers side activities	Push Commits, Close PR, Clean Unused Branches	All Teams			
19		Deployment Related Changes	Pause Builds or any automation related Tasks	Kraken		Buildkite, Terraform Cloud, TeamCity	
20		Bitbucket	Freeze all repositories so that no one should push new code changes during the migration	Kraken			One way to freeze Bitbucket is rename Workspace ID/Slug (Tested)
21		Bitbucket	Take full backup of Bitbucket repositories	Kraken		TODO How to take complete backup, check with team	
22	**Mini Migration**	Non Critical Repositories	Verify Migration check if we can do it more efficiently Update script If required Identify Challenges	Kraken			
23	Migration	Migration	Initiate Migration	Kraken		Setup Migration Environment Create Token in GitHub, Repository Administrator, Webhook, Organization Read permissions Scrips are ready, Any changes in repository settings, scripts needs to be updated	
24	Post Migration	Deployment Related Changes	Buildkite	Kraken		Add SSH key in GitHub which is used with Bitbucket Connect GitHub to Buildkite Change repository Provider to GitHub for all pipelines : script are ready Test Pipeline	
25		Deployment Related Changes	TeamCity	Kraken		Create backup of TeamCity server and database Create token on GitHub and connect TeamCity to GitHub Update Parameters in TeamCity TODO SSH key is use "TeamCity to Bitbucket Cloud SSH Key" check with Team	Some common Parameters, we need to update in root project gitUserName gitPassword BitbucketCloudUsername BitbucketCloudPassword git.password git.username In Bitbucket organization Name is replace me-git, In TeamCity We need to change all References to GitHub replace me Organization name
26		Deployment Related Changes	Update Terraform Module References	Kraken		TeamCity pipeline has been created for this run the pipeline and check if endpoints updated if required update pipeline configuration	Terraform environment terraform-core-infrastructure terraform-botium terraform-tfcloud-environment terraform-pulsarcx infra-scripts terraform-iam terraform-modules
27		Deployment Related Changes	Terraform Cloud	Kraken		Add/update SSH key GitHub/TF cloud Change VCS Provider to GitHub update TeamCity terraform cloud pipeline Update terraform cloud agent scripts	
28		Snyk	Snyk Integration with GitHub			Import project into snyk if required R&D is their any plugin available for snyk integration Reintegrate modernization project into snyk	Non blocker post migration activity
29	Post Migration Verification	Deployment Related Changes	Restart Builds or any automation related Tasks and Monitor Builds	Kraken		Buildkite, Terraform Cloud, TeamCity	
30		Verification	Test if builds are successful			Buildkite, Terraform Cloud, TeamCity	
31		Developers side activities	Local environment setup	All Teams		To Update Existing Local Repositories, Use the following commands to update the remote origin URL from Bitbucket to GitHub git remote set-url origin <GitHub Repository URL> git pull origin <branch-name> or git fetch origin	