Plan

Document Level Classification

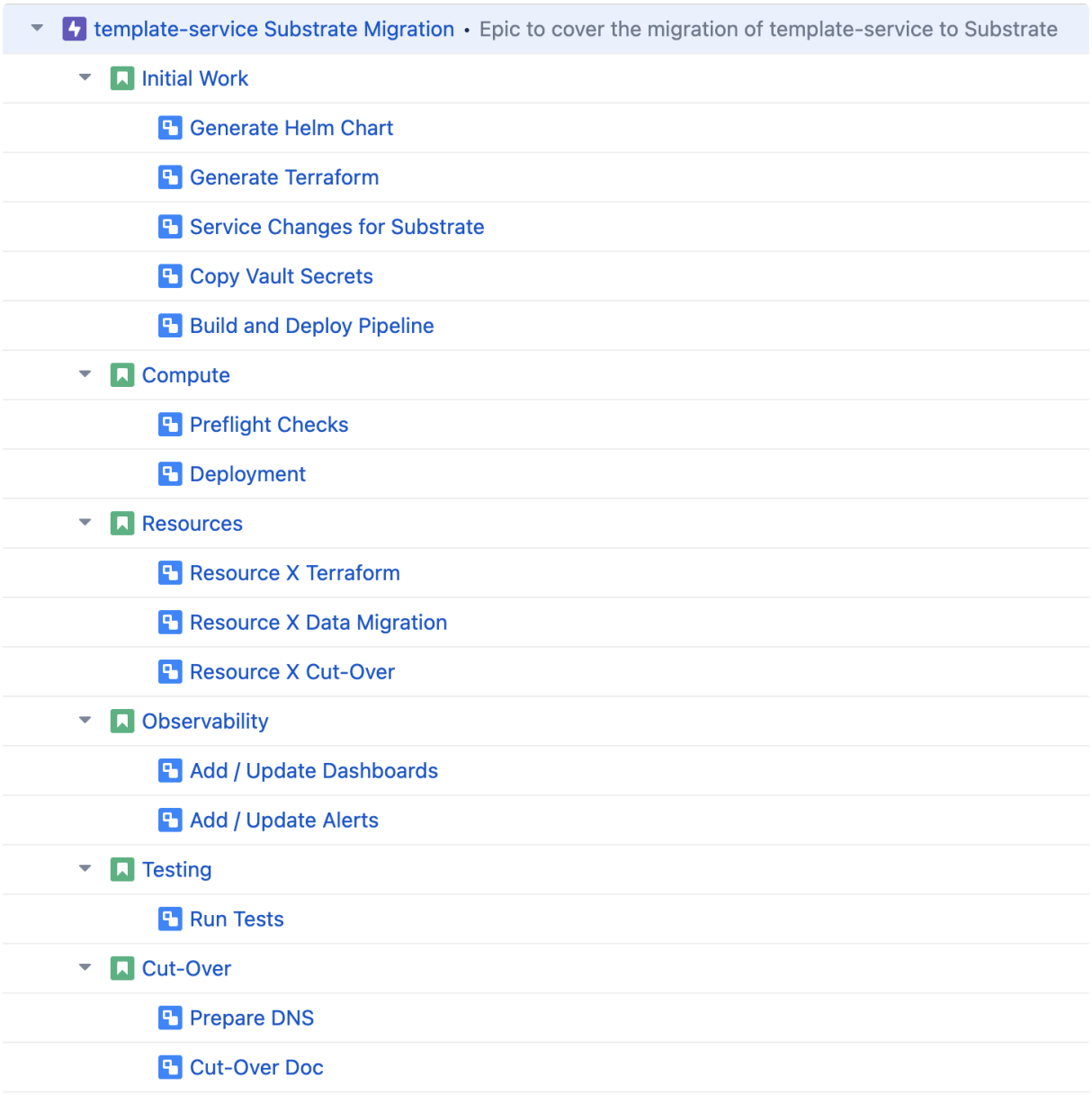
200

    - Rollback
- Outcomes

Mosaic macros cannot be exported to this format.

Now that you have gathered the information around your service, you will need to determine a few areas of planning. These will allow you to organize, communicate and schedule work.

Project Management and Visibility



We suggest organizing the work into a JIRA Epic that reflects the below described plan. From this structure you can create additional tickets if desired and link them to the these tickets.

- Start by creating an Epic ticket in your teams JIRA project.
- Add Stories for:
    - Initial Work
    - Compute
    - Resources
    - Observability
    - Testing
    - Cut-Over

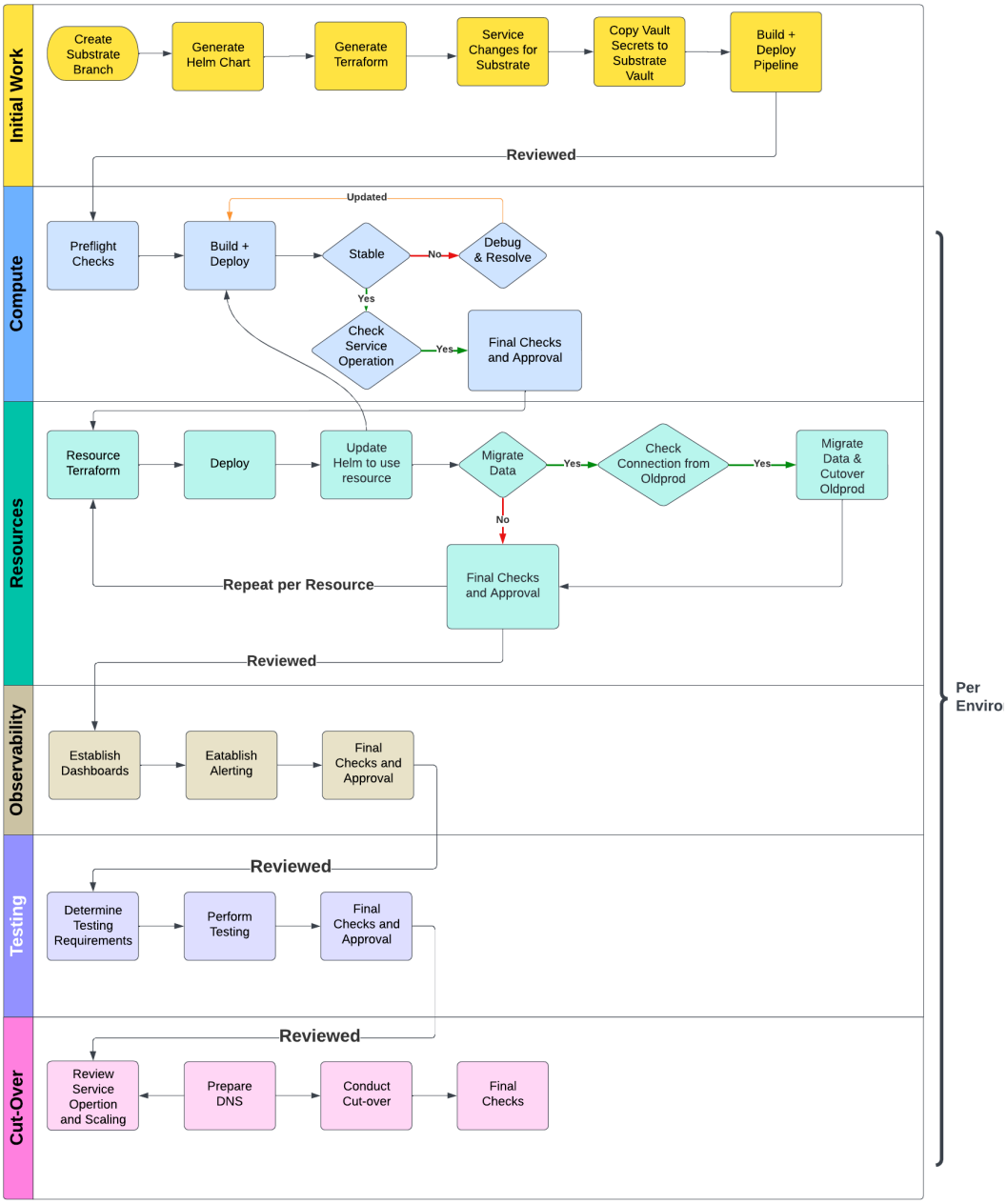
As you progress through the next section of planning, add sub-tickets to these Story tickets and add details of what you need to work on.

For example, under Initial Work you may have the following sub-tasks:

- Generate Helm Chart
- Generate Terraform
- Service Changes for Substrate
- Copy Vault Secrets
- Build and Deploy Pipelines

📝 Put a link to the JIRA Epic Ticket in your Migration Worksheet (cell B8). The review group will use this as part of the assessment.

Workload Migration Plan



You should begin to put together a plan of how you will migrate your service to Substrate as this will make it easier to divide work and order work appropriately. To help you get started we have provided the below approach by grouping tasks into blocks of work. This is a guide and each service may have their own unique cases, but it is enough to at least help you determine where certain edge cases should go. Each block of work should be reviewed before beginning the next block of work as in most cases, blocks of work will be dependent on the previous block.

Initial Work

This is where you will begin your migration work. At this point you should already have a dev and live substrate account for use.

Create Substrate Branch

By creating a branch for all your migration work, you can prevent any issues with oldprod deployments. Here you will make any application code changes and add your Helm Charts and Terraform for Substrate.

Generate Helm Chart

To get your application running in Substrates' Kubernetes Cluster, you will need a Helm chart based on epic-app to run your service. The Inflow tool allows you to generate this chart for your service. You can learn more about the Inflow Helm Generator in the UI: https://inflow.ol.epicgames.net/tool/helm (click on for the guide).

Generate Terraform

Initially you will only need barebones Terraform with a Security Group for your load balancers. As you move onto migrating other resources, you can add your Terraform here.

Service Changes for Substrate

In order to get your service to boot in Substrate, you most likely need to make some application code changes to account for the difference in underlying infrastructure. The commo changes are:

1. Change logging configuration to send logs to stdout [LINK TO GUIDE]
2. Change application code to account for any EC2 metadata fetching [LINK TO GUIDE]
3. IAM accounts [LINK TO GUIDE]

Copy Vault Secrets to Substrate Vault

In order for your service to access vault secrets, it will need to access them from Substrate Vault (a separate Vault from oldprod). We have provided a guide on how to do so. [LINK TO GUIDE]

Build and Deploy Pipeline

You will need a separate build and deploy pipeline for Substrate. You can choose to continue using whichever tool your team currently uses or you can use this opportunity to move to another platform. Create these pipelines and set them to run builds from your new branch. Templates for Github Actions and Codefresh are provided to get you started: [LINK TO GUIDE]

Compute

After the initial work, you should be in a position to build and deploy your application to Substrate where you can begin to debug issues and test your application. Initially try get one pod to boot, then scale out. When it comes to cut-over of traffic, it’s useful to match scale with oldprod and then start optimizing.

Preflight Checks

The first boot of your service in Substrate is the most difficult as you need to ensure your service can connect to other services and to resources still in oldprod. To help make the application boot and run, we suggest running a set of preflight checks per environment to ensure your application has connectivity to oldprod resources, services and services in other substrate accounts. We have provided a tool to check connectivity to these services and resources, sourcing the configuration information from both the Helm chart values file and the migration worksheet. Use the tool to help test connectivity and resolve any connectivity issues before deploying your application. [LINK TO TOOL]

Build and Deploy

With connectivity checked, you can trigger a build of your substrate branch and a deployment of this new build. After a deployment you should check your deployment is stable and resolve any issues if it is not. Repeat this process until the deployment is stable and confirm by checking connectivity to the application through the Ingress endpoint.

Final Checks and Approval

Once you have your application deploying to Substrate, it is operational, stable and can be connected to, you are in a good position to progress to moving over resources to your substrate account.

Resources

Now that your application is running in Substrate you can begin to migrate any resources you highlighted in the Assessment phase. As you will be touching resources that oldprod depends upon, exercise caution here as there is potential to impact services.

The general approach for a resource is to stand-up the new resource in your substrate account with Terraform, reconfigure your Helm values file to use that resource and redeploy your service in Substrate.

If you are migrating data you will need to consider a cut-over strategy that will support running your service in oldprod and substrate in parallel. [LINK TO DOCS]

Observability

Setup new or ensure existing dashboards are using Substrate metrics. Likewise setup alerting to use Substrate metrics as the datasource. If you haven’t already, this is a good opportunity to move your alerts to Terraform.

Testing

Although you will be testing your service during the compute and resource phases, at this point we want to perform some end-to-end testing to ensure everything is operating as expected. This will be different for each team and service but some suggestions are:

- Asking other service teams to test their service that depends on yours - they will need to change their configuration to point at your service in Substrate
- Running integration tests against your service in Substrate
- Running load tests against your service in Substrate

Determine what tools you have available to you in order to test your service.

Cut-Over

This guide focuses on cut-over topics when approaching critical environments such as live and/or prod, but the approach can be applied to lower environments, and should be if those lower environments are also critical (such as gamedev). The actual cut-over work will be conducted later on and you can submit your cut-over plan documentation for review close to the date.

Use this opportunity to become familiar with where your cut-over will need to take place. These are commonly:

- Changing vanity DNS records (ol.epicgames.com / ol.epicgames.net) to point to Substrate (directly or weighted)
- Changing CDNs to point to substrate

At the bottom of the Environment sheets in your Worksheet you will see the DNS entries for that environment and how they are mapped and where those DNS entries are configured. These will help you plan what DNS needs to be changed.

Consider how you will cut-over:

- Weighted DNS to slowly shift traffic to Substrate
- Hard DNS cutover
- Is a downtime required

A note on environment rollout: The above work should be repeated per environment. It is at your discretion whether you do this sequentially or in parallel (or some back and forth between environments). We recommend getting a lower environment to the point of your application deploying and stable before replicating this work to other environments as this is where you will discover the main gotchas of your service in Substrate.

Communication Guidelines

Any work that has the potential to take a service offline should be communicated to the relevant stakeholders with timely notice. Examples of when to communicate are:

1. Switching datastores from oldprod to substrate
2. Routing requests to substrate.
3. Changes to access controls (Security Groups, ALBs, CDNs, etc)
4. Any ongoing issues in Substrate or Oldprod due to changes.

When communicating with stakeholders, details worth sharing include:

1. What you’re planning to do - database move, CDN change, etc
2. What to expect - will there be a downtime, will it cause the service to run slow while cache warms, or will this go unnoticed?
3. Timeline of events
4. Risks and Challenges - outline what could go wrong and what to look out for.
5. Teams and Ownership - who is responsible for work

6. Communication Channels - where to follow along or where to view or report issues.

Effective communication is about sharing updates even if people are following along:

1. Before you're about to perform work (a week, a day, an hour before)
2. When work has begun
    1. Update if you have hit an issue and are investigating
2. When work has finished
    1. Successfully completed
    2. Rolled back and will retry at a later time

An example of a message you might share:

Hi everyone - we in the MegaGamingCreator team will be migrating wonderful-service to Substrate next week at 12 EST. To do this we will change the DNS records wonderful-service-public.ol.epicgames.com to point to the Substrate load balancers. This will be a hard cut-over. We don't expect any issues but it is possible the service will run slower due to cache warming. You can follow the progress in #megagamingcreator-ext where we will share updates. If you notice any issues please reach out to us there.

To make it easier, it is worth preparing some text now as a template that you can use.

Who do you communicate with? Anybody who may be affected by your work. We also suggest including LiveOps as they can help coordinate if issues arise.

- LiveOps
- Your team
- The team's your service depends on - these are listed under "Incoming Services - Internal" in the worksheet but there may be others.

Take the time now to identify the stakeholders and their communication channels. This may be through e-mail, Slack, Sideband or any other means. The worksheet will help you identify service owners that will need to be communicated with.

Timing Work

Timing is very important - migration work can potentially become disruptful to not only live production services, but also to internal dev environments, in particular gamedev. This may be the case when you start migrating dev environment traffic from oldprod to substrate and issues go unnoticed in that environment.

Therefore along with communicating work with stakeholders, it is also important to pay attention to scheduled events for Epic, such as MegaSales, Free Game give aways, Fortnite Events, and anything else that draws a lot of traffic.

Doing a migration of traffic in the days, weeks, and even month leading to these events carries risk. Even if the migration of traffic goes successfully, other unknowns may appear shortly afterwards, such as performance issues or access issues not caught in time.

Check the #lops-event-planning-ext slack for updates on event timelines.

Rollback

As you progress through planning your migration, you should consider an approach for rollback during the different phases. For example, if you are migrating a datastore to Substrate, consider what you can do if the migration does not go as planned and you need to switch back to the oldprod datastore.

Likewise the cut-over of traffic (discussed below) should contain a rollback strategy if issues occur in Substrate and traffic must be switched back to oldprod.

Outcomes

At the end of section your Migration Worksheet document should now be filled in with the information you've gathered and reviewed.

At this point you can submit the Migration spreadsheet for review by the review group if you wish to available of the service. This will help ensure any potential issues are identified.

Mosaic macros cannot be exported to this format.