Creating Basemaps

1. **Key Components**

2. **Procedural Outline**

   2.1. Obtain Data

   2.2. Clean/Prepare Data

   2.3. Declare Parameters

   2.4. Styling

   2.5. Labeling

   2.6. SLD Creation

   2.7. Geoserver Uploads

   2.8. Edit SLD Styling

   2.9. SLD Labeling

Key Components of Basemap Styling

1. **Scale-dependant styling**
Most Web maps allow zoom control, as a result, maps must be styled appropriately at each zoom level.
2. **Dynamic Labeling**
Labels must be responsive to changes in map extent and zoom level.

Creating a basemap is like creating a bunch of different maps, one for each zoom level, and then allowing users to seamlessly navigate them.
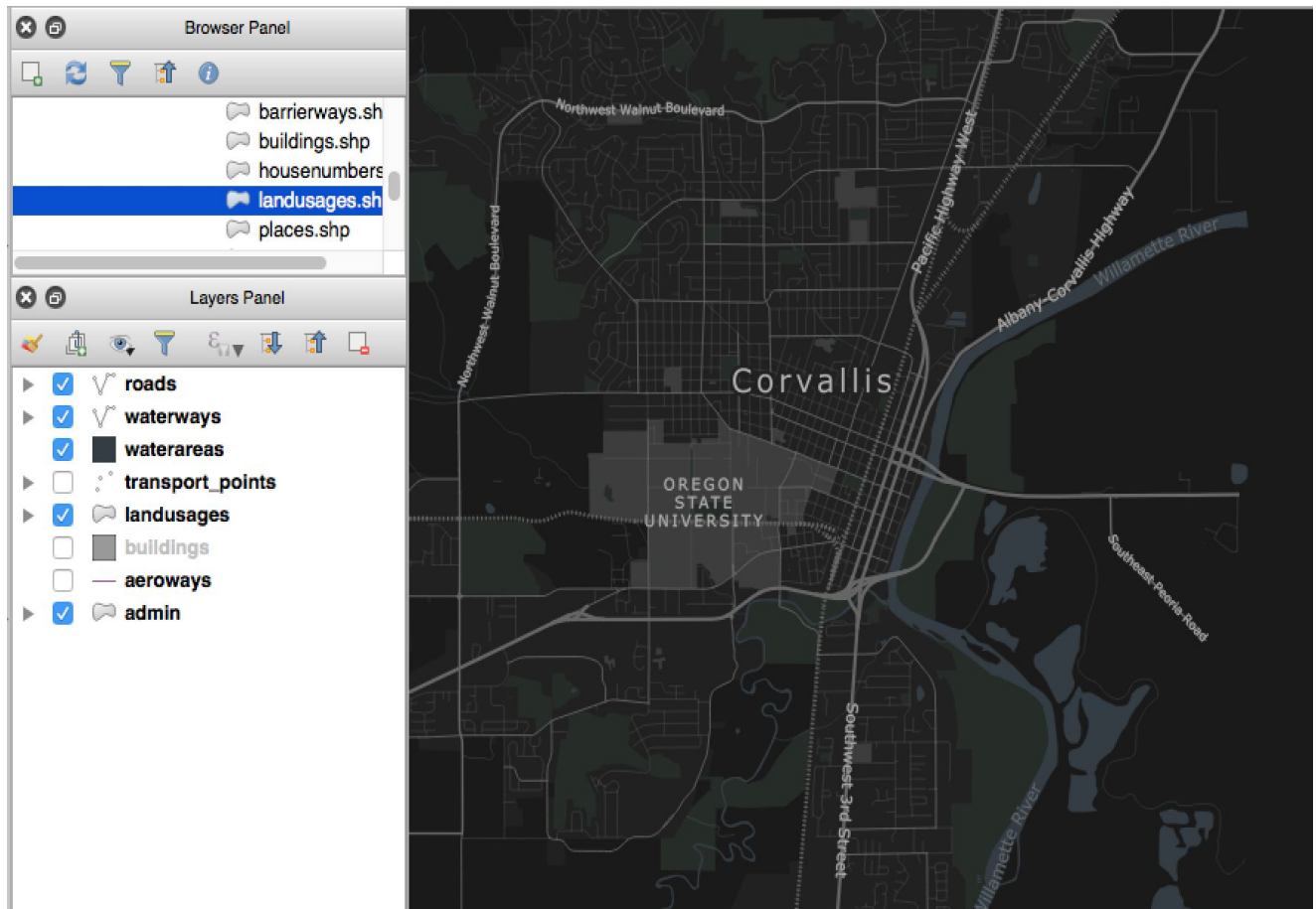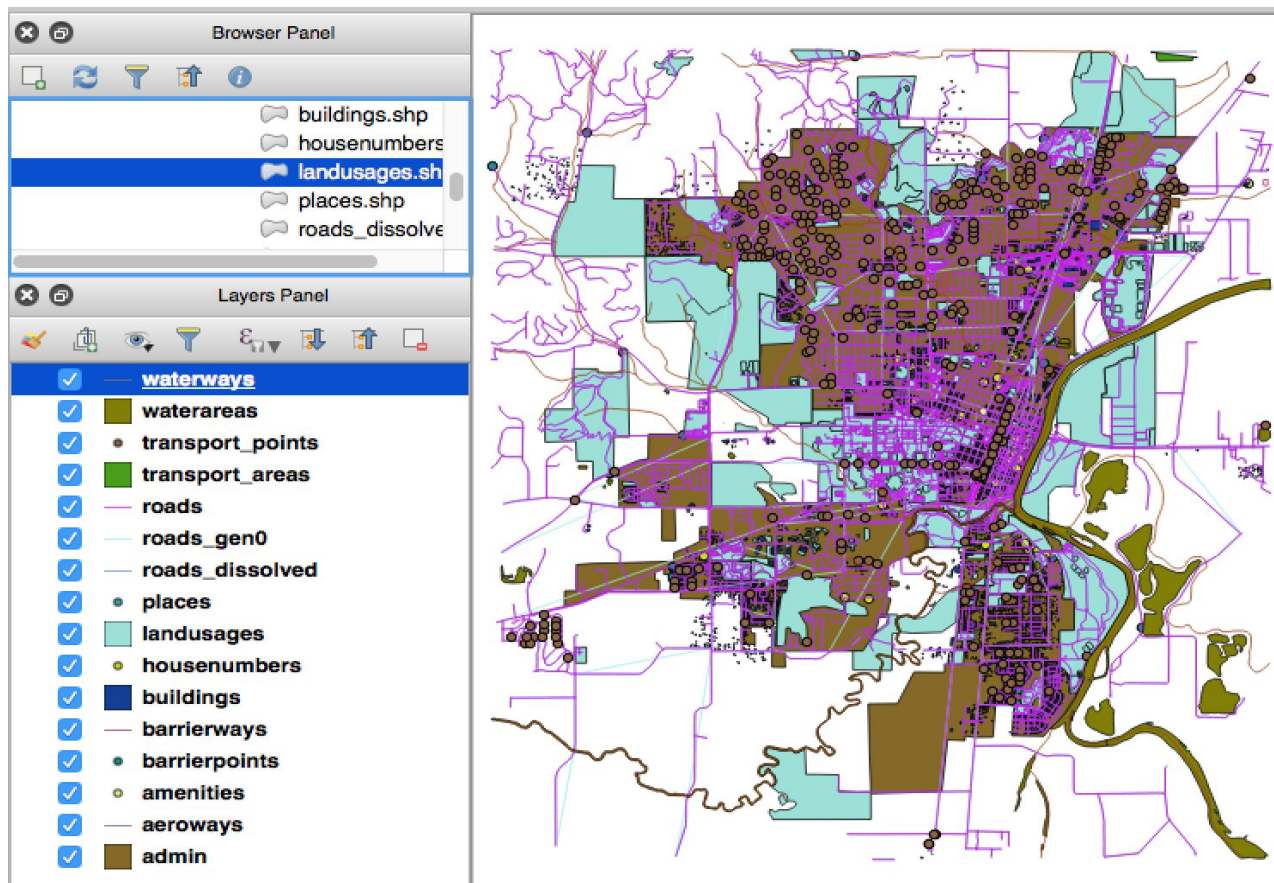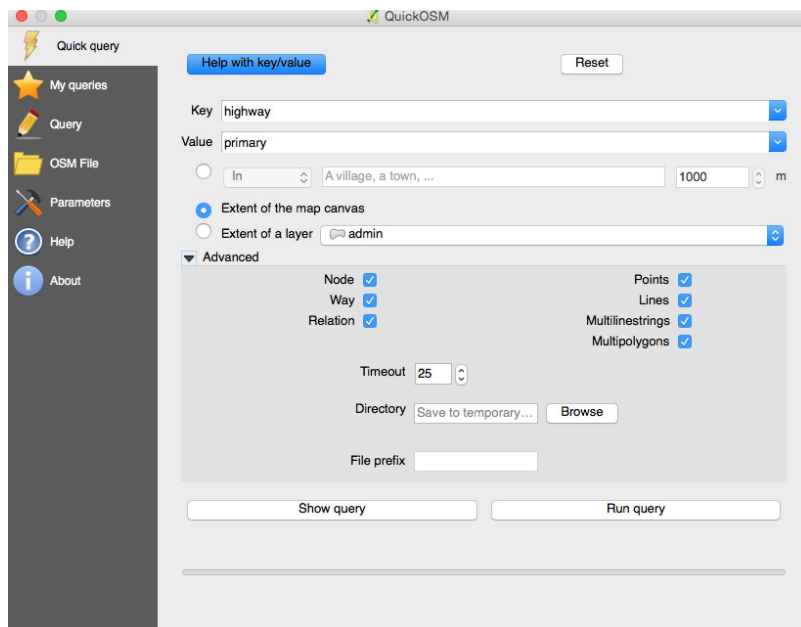
Step 1: Obtain Data

Basemaps show a lot of geographic features and therefore require a lot of different data.

Ways to obtain basemap data:
1. QuickOSM Plugin - QGIS

2. MapZen
3. Oregon Spatial Data Libarary/Data.gov

Step 2: Clean/Prepare Data

Any data you obtain from basemaps likely has some extraneous features or layers than can be omitted.

There is also an element of creative choice, as you get to decide which data, or parts or data are important enough to warrant symbolization.

For example, if I pull data from MapZen and there's a layer that has traffic circles, I need to decide if that's something I'm actually going to symbolize on my map.

Step 2: Clean/Prepare Data

✓	— waterways
✓	▢ waterareas
✓	• transport_points
✓	▢ transport_areas
✓	— roads
✓	— roads_gen0
✓	— roads_dissolved
✓	• places
✓	▢ landusages
✓	• housenumbers
✓	▢ buildings
✓	— barrierways
✓	• barrierpoints
✓	• amenities
✓	— aeroways
✓	▢ admin

What spatial features are actually stored in each of these layers? What do you think is necessary? What isn't?

Step 3: Declare Parameters

Before you begin styling, it is important to consider the minimum and maximum zoom levels that you want your basemap to cover. For the purposes of this lab, I chose to use zoom levels 12-15. You may choose differently.

We do this ahead of time so we don't have to style layers for zoom levels we know we'll never use.

Step 3: Declare Parameters

Level of Detail	Map Width and Height (pixels)	Ground Resolution (meters / pixel)	Map Scale(at 96 dpi)
1	512	78,271.5170	1 : 295,829,355.45
2	1,024	39,135.7585	1 : 147,914,677.73
3	2,048	19,567.8792	1 : 73,957,338.86
4	4,096	9,783.9396	1 : 36,978,669.43
5	8,192	4,891.9698	1 : 18,489,334.72
6	16,384	2,445.9849	1 : 9,244,667.36
7	32,768	1,222.9925	1 : 4,622,333.68
8	65,536	611.4962	1 : 2,311,166.84
9	131,072	305.7481	1 : 1,155,583.42
10	262,144	152.8741	1 : 577,791.71
11	524,288	76.4370	1 : 288,895.85
12	1,048,576	38.2185	1 : 144,447.93
13	2,097,152	19.1093	1 : 72,223.96
14	4,194,304	9.5546	1 : 36,111.98
15	8,388,608	4.7773	1 : 18,055.99
16	16,777,216	2.3887	1 : 9,028.00
17	33,554,432	1.1943	1 : 4,514.00
18	67,108,864	0.5972	1 : 2,257.00
19	134,217,728	0.2986	1 : 1,128.50
20	268,435,456	0.1493	1 : 564.25
21	536,870,912	0.0746	1 : 282.12
22	1,073,741,824	0.0373	1 : 141.06
23	2,147,483,648	0.0187	1 : 70.53

Step 4: Styling

Styling a basemap is different than styling a static map, but it can be thought of as styling a series of connected static maps.
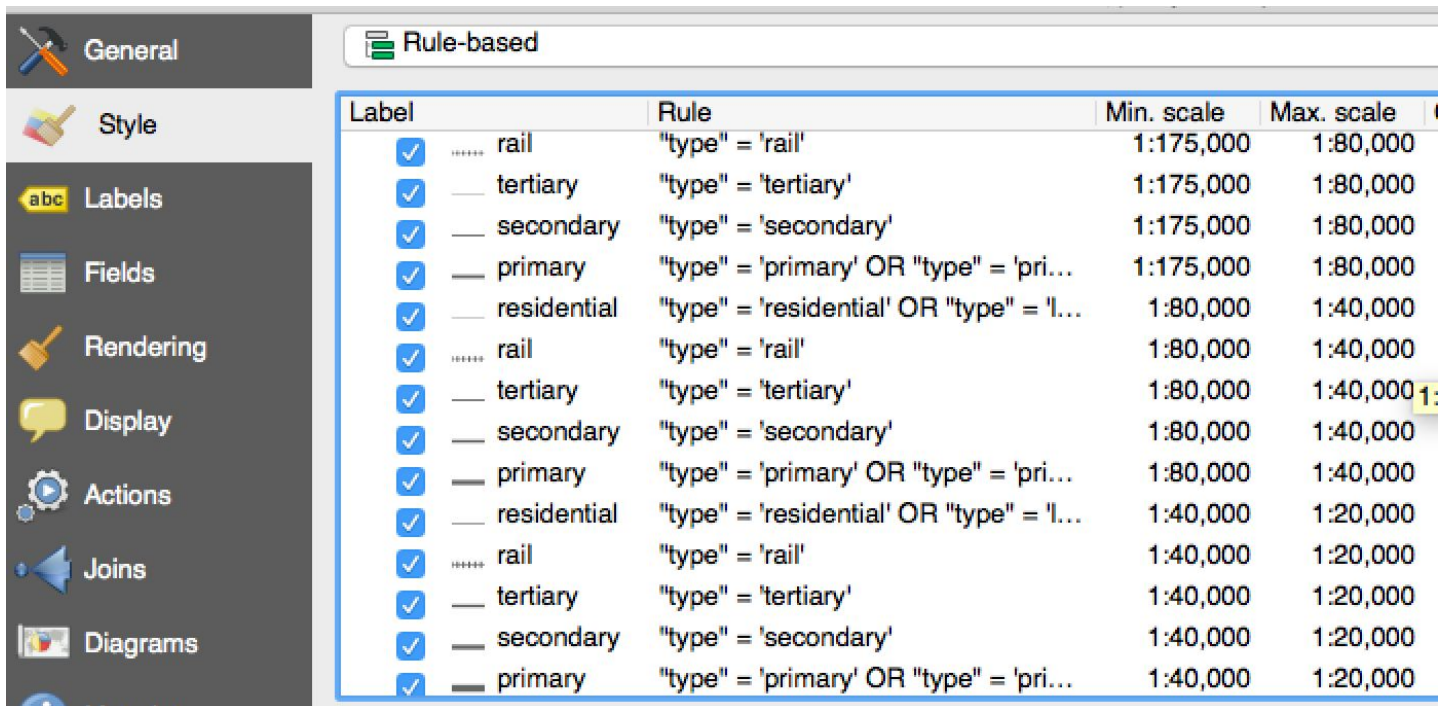
Rule-Based styling in QGIS is going to be very useful to you, as well as scale-dependant styling. If you're styling roads, you need to classify roads at each zoom level and style them accordingly.

Step 4: Styling

Here's a sample styling workflow:
- Before you begin, make sure your units are in points, this will make SLD transfer easier.
- Begin at smallest scale/zoom
  - Decide which layer is most spatially prominent at that scale
    - Classify hierarchical data if necessary
    - Declare min/max scale for each classification
  - Repeat for other features, until all desired features at that zoom level are styled.
  - Repeat for other zoom levels, until all desired zoom levels are featured

Step 4: Styling

Label		Rule	Min. scale	Max. scale	
☑	rail	"type" = 'rail'	1:175,000	1:80,000	
☑	tertiary	"type" = 'tertiary'	1:175,000	1:80,000	
☑	secondary	"type" = 'secondary'	1:175,000	1:80,000	
☑	primary	"type" = 'primary' OR "type" = 'pri...	1:175,000	1:80,000	
☑	residential	"type" = 'residential' OR "type" = 'l...	1:80,000	1:40,000	
☑	rail	"type" = 'rail'	1:80,000	1:40,000	
☑	tertiary	"type" = 'tertiary'	1:80,000	1:40,000	1:
☑	secondary	"type" = 'secondary'	1:80,000	1:40,000	
☑	primary	"type" = 'primary' OR "type" = 'pri...	1:80,000	1:40,000	
☑	residential	"type" = 'residential' OR "type" = 'l...	1:40,000	1:20,000	
☑	rail	"type" = 'rail'	1:40,000	1:20,000	
☑	tertiary	"type" = 'tertiary'	1:40,000	1:20,000	
☑	secondary	"type" = 'secondary'	1:40,000	1:20,000	
☑	primary	"type" = 'primary' OR "type" = 'pri...	1:40,000	1:20,000	

Left panel: General, Style, Labels, Fields, Rendering, Display, Actions, Joins, Diagrams. Top: Rule-based

Shown above is sample styling for a roads layer. Here, roads are classified based on the "type" attribute and then organized based on their minimum and maximum scales.

Note that the minimum and maximum scales don't correspond directly with zoom levels. Exact zoom level scale varies with computer resolution, it is therefore easier to set the min/max scale so that the desired zoom level is somewhere in the middle. When you're styling, view your map at specific zoom levels.

Step 5: Labeling

Labels in QGIS can't be directly exported to SLD files, but using QGIS to prepare desired labeling will make creating SLD labels in geoserver much easier.
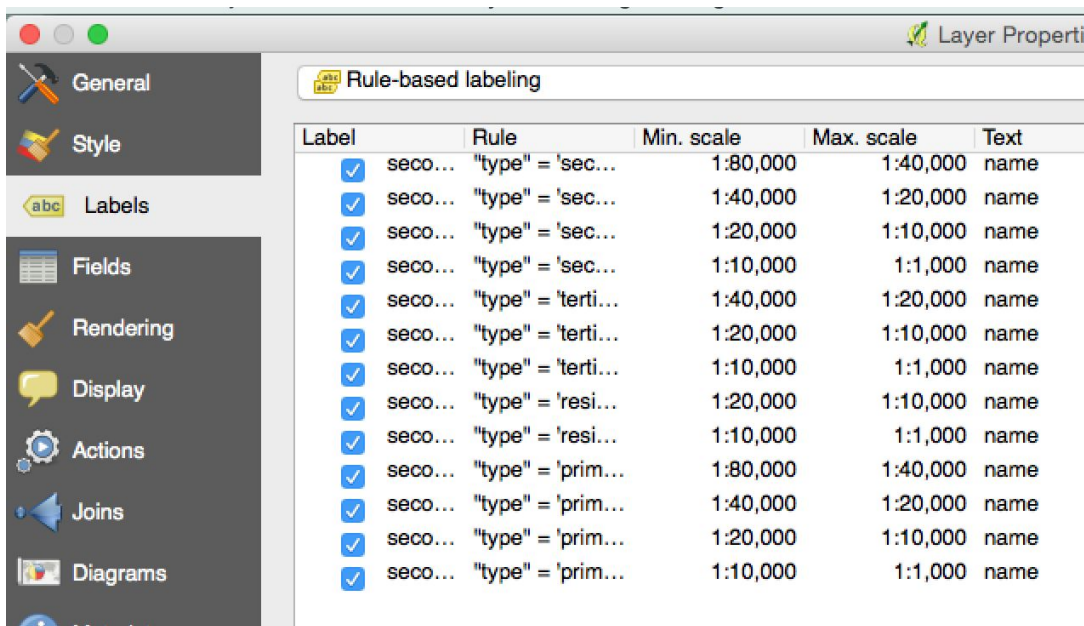
Using rule-based labeling is very useful. However, it is important that your labeling rules align with your styling rules. When you export to SLD, you'll only be able to label based on your styling rules.

Step 6: SLD Creation

After you're finished styling, you need to export all of your styles to SLD files so that they can be used in Geoserver.
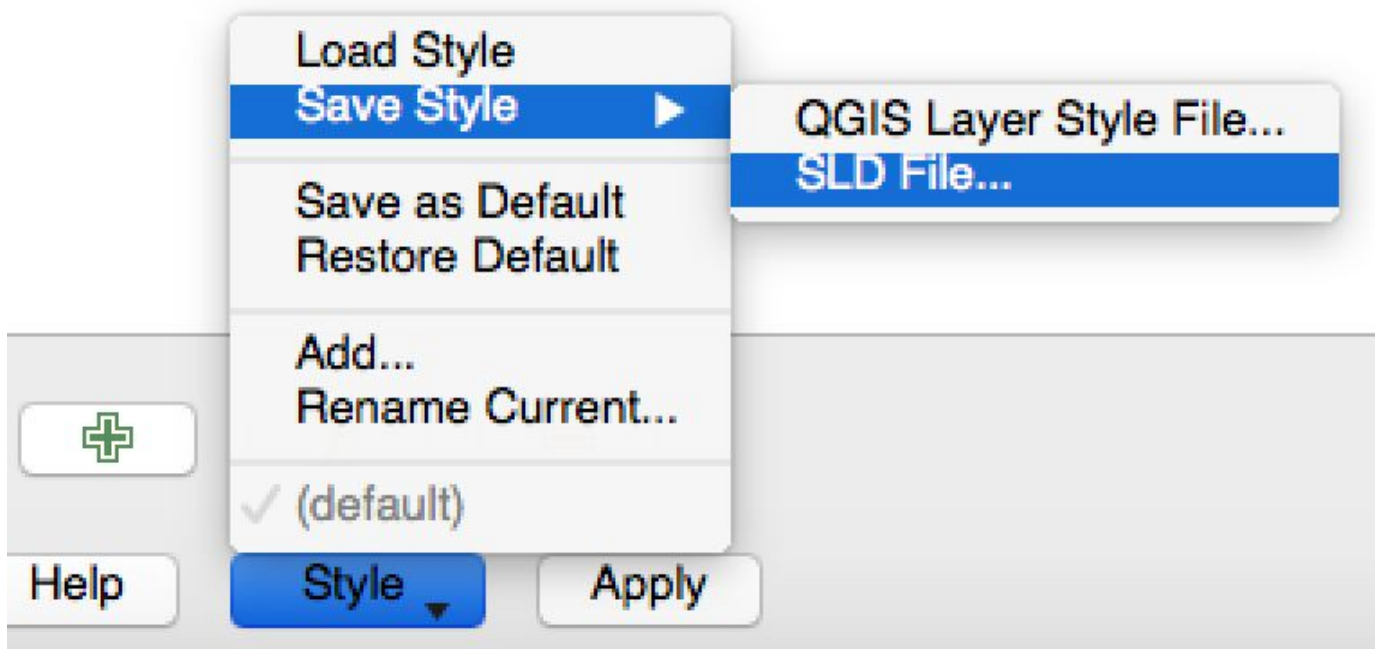
At the bottom of the 'Style' tab in the layer properties window in QGIS, there's an option to save style as an SLD file.

Step 7: Upload to Geoserver

Once work in QGIS is done, all the component parts must be combined in Geoserver in order to be made accessible for use in web maps.

Step 7: Upload to Geoserver

Workflow:
- Create a basemap workspace
- Upload all data files into a store within the workspace
- Upload all SLD files as styles
- Publish layers with the appropriate stylesheets
- Preview layers
- Combine layers in layer group for basemap
- Preview Layer Group

Step 7: Upload to Geoserver

Once everything has been uploaded, you can view your basemap in Leaflet.

```
24
25  ▼  <body>
26
27         <div id="map"></div>
28
29
30  ▼  <script>
31
32         var map = L.map('map',{scrollWheelZoom:true}).setView([44.5645, -123.2620],12);
33
34  ▼     L.tileLayer.wms("http://mapio.us/geoserver/baldricg/wms", {
35             layers: 'baldricg:corvallis_basemap',
36             format: 'image/png',
37             minZoom: 12,
38             maxZoom: 15,
39             transparent: true,
40             attribution: "Basemap created by Gareth Baldrica-Franklin"
41  ⌐     }).addTo(map);
42
43
44
45
46  ⌐  </script>
47
48  ⌐  </body>
49
```

Step 8: Edit SLD Styling

Once you create a layer group for your basemap, you may realize that your web styling doesn't exactly match the styling you did in QGIS.

These can be edited manually in the SLD style window in Geoserver.

Things to pay attention to:
- Stroke Size
- Presence of strokes around polygons
- Specific coloration

Step 9: SLD Labeling

Using SLD files to label is very tedious, and must be done manually.

Two resources on how to do so are:

SLD Cookbook:
http://docs.geoserver.org/stable/en/user/styling/sld/cookbook/lines.html#optimized-label-placement

Geoserver Documentation:
http://docs.geoserver.org/latest/en/user/styling/sld/reference/labeling.html

Step 9: SLD Labeling

One of the most important sets of features that will be used when SLD labeling is the "VendorOption" tag. This will allow for more dynamic placement options.

```xml
241    <se:TextSymbolizer>
242      <se:Label>
243          <ogc:PropertyName>name</ogc:PropertyName>
244      </se:Label>
245      <se:Font>
246          <se:SvgParameter name="font-family">Arial</se:SvgParameter>
247          <se:SvgParameter name="font-size">8</se:SvgParameter>
248          <se:SvgParameter name="font-style">normal</se:SvgParameter>
249      </se:Font>
250      <se:LabelPlacement>
251          <se:LinePlacement />
252      </se:LabelPlacement>
253      <se:Halo>
254          <se:Radius>0.8</se:Radius>
255          <se:Fill>
256              <se:SvgParameter name="fill">#212121</se:SvgParameter>
257          </se:Fill>
258      </se:Halo>
259      <se:Fill>
260          <se:SvgParameter name="fill">#b2b2b2</se:SvgParameter>
261      </se:Fill>
262      <se:VendorOption name="followLine">true</se:VendorOption>
263    </se:TextSymbolizer>
```

You will have to copy code like this into the SLD for each rule you want to include labels for in your styling.